

**Pekka Mäkiaho and Timo Poranen (eds.)**

# **Software projects 2013-2014**



UNIVERSITY OF TAMPERE  
SCHOOL OF INFORMATION SCIENCES  
REPORTS IN INFORMATION SCIENCES 27

TAMPERE 2014

UNIVERSITY OF TAMPERE  
SCHOOL OF INFORMATION SCIENCES  
REPORTS IN INFORMATION SCIENCES 27  
MAY 2014

**Pekka Mäkiaho and Timo Poranen (eds.)**

# **Software projects 2013-2014**

SCHOOL OF INFORMATION SCIENCES  
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-9481-9

ISSN-L 1799-8158  
ISSN 1799-8158

## Preface

This report contains stories of 13 software development projects from academic year 2013-2014. The students came from Project Work (TIEA4) and Software Project Management – Practice (TIETS19) courses. The stories describe what kind of experiences groups got during the project and what was the software product that came out from the project. In the end of each story there are project statistics.

**Table 1: General project statistics.**

Project	Type	Client	Dev. Mod.	Group	Hours
Itikka	WWW	University	Scrum	3+3	820
Field Optimization	WWW	Company	Scrum	2+4	1190
PMO	WWW	Association	Scrum	3+3	1247
math.fi	WWW	Association	Scrum	3+4	1250
Pingismaisteri	WWW	Association	Scrum	3+3	906
Elementary mathematics	Kinect	University	Scrum	3+4	953
Time Distance Speed	Kinect	University	Scrum	4+4	1167
Puzzle game	Android	University	Incr.	3+4	1000
Disc Golf	Android	University	Scrum	3+2	932
Seek'n'Share	WWW	University	Scrum	4+4	1245
Flowing Hours	WWW	Company	Ad hoc	3+4	810
GroService	WWW+Android	Company	Scrum	4+4	1049
Höntsy	WWW	Company	Scrum	3+3	918

Table 1 gives an overview of the projects. For each project, there is project type (WWW, Android or Kinect), client (University, Company or Association), development model, group size (number of managers + number of developers) and working hours of the project.

Although 11 projects applied Scrum (“Scrum-but”) development model, there were usually some differences when compared to standard Scrum: no face-to-face daily meetings, no fixed length iterations, many managers in the team, no sprint retrospective...

Table 2 contains general course statistics (number of projects and usability teams, number of students in the courses and average project size in working hours) starting from year 2005.

**Table 2: Course statistics 2005-2014.**

Academic year	Projects	Usability teams	PW students	SPM students	Average project size
2005-6	19	1	98	8	1008
2006-7	18	2	87	34	1089
2007-8	14	1	70	29	997
2008-9	10	1	60	39	1643
2009-10	15	1	80	34	1151
2010-11	13	1	70	27	1230
2011-12	14	0	67	30	1331
2012-13	13	0	54	39	1109
2013-14	13	0	46	41	1038

School of Information Sciences (SIS) offered version control services (subversion) and Redmine-project management tool for all projects. Balsamiq was used in many projects for user interface design. SIS's projectWiki was used to maintain course and project related documentation and guidelines: <https://projectwiki.sis.uta.fi>. The wiki also contains some articles on project management and project management tools, including lists of end-products currently in use, course related publications and course related videos:

- [https://projectwiki.sis.uta.fi/wiki/Finished\\_projects](https://projectwiki.sis.uta.fi/wiki/Finished_projects)
- [https://projectwiki.sis.uta.fi/wiki/Course\\_publications](https://projectwiki.sis.uta.fi/wiki/Course_publications)
- [https://projectwiki.sis.uta.fi/wiki/List\\_of\\_project\\_videos](https://projectwiki.sis.uta.fi/wiki/List_of_project_videos)

Course staff thanks our clients and students for great projects!

Pekka Mäkiäho and Timo Poranen

Tampere, May 2014

Preface .....	i
IT-ympäristön konfiguraation hallinnan järjestelmä.....	1
Field Optimization Tool .....	7
Pirkanmaan musiikkiopisto.....	17
Math.fi .....	22
Pingismaisteri.....	27
Games for Elementary Mathematics Learning Environment .....	43
Time Distance Speed experiment for High School Students.....	59
Android Puzzle Game.....	67
Disc Golf Game Project.....	74
Seek N Share.....	78
Flowing Hours.....	84
GroSerwise Systems.....	92
Höntsyt.....	101

# IT-ympäristön konfiguraation hallinnan järjestelmä

## Overview

Project was about creating a system for IT- configuration management (CMDB) for the customer. The customer was looking for a simple solution / prototype for a CMDB-type tool, that they could then continue or use as basis for evaluation other software.

The main idea behind the customer need is the fact that an IT-infrastructure contains vast number of objects and relations that need to be managed. For example, when a component is changed, it's important to know what it affected, and also, in error cases, when changes were made to a component.

Customer: UTA IT-administration.

The application is a web-based tool, in which the user can create Types (for example Router / Computer / Application Server / Database), and then create instances of these types. For example "Cisco Router CS101" could be an instance (item) of router (type).

Furthermore the tool provides the possibility to link items to each other via Relationships, this enables the user to see which items are linked together, thus providing the user with a tool for understanding how changes might affect components in the network.

The UI is based on web elements and adding and modifying of items/types is done with simple form-like pages.

Home > Items > Add item

### Add item

Type \* Application server

Name \*  Name cannot be blank.

Operating system \*

Purpose

Warranty date mm/dd/yyyy

Add

### Relationships

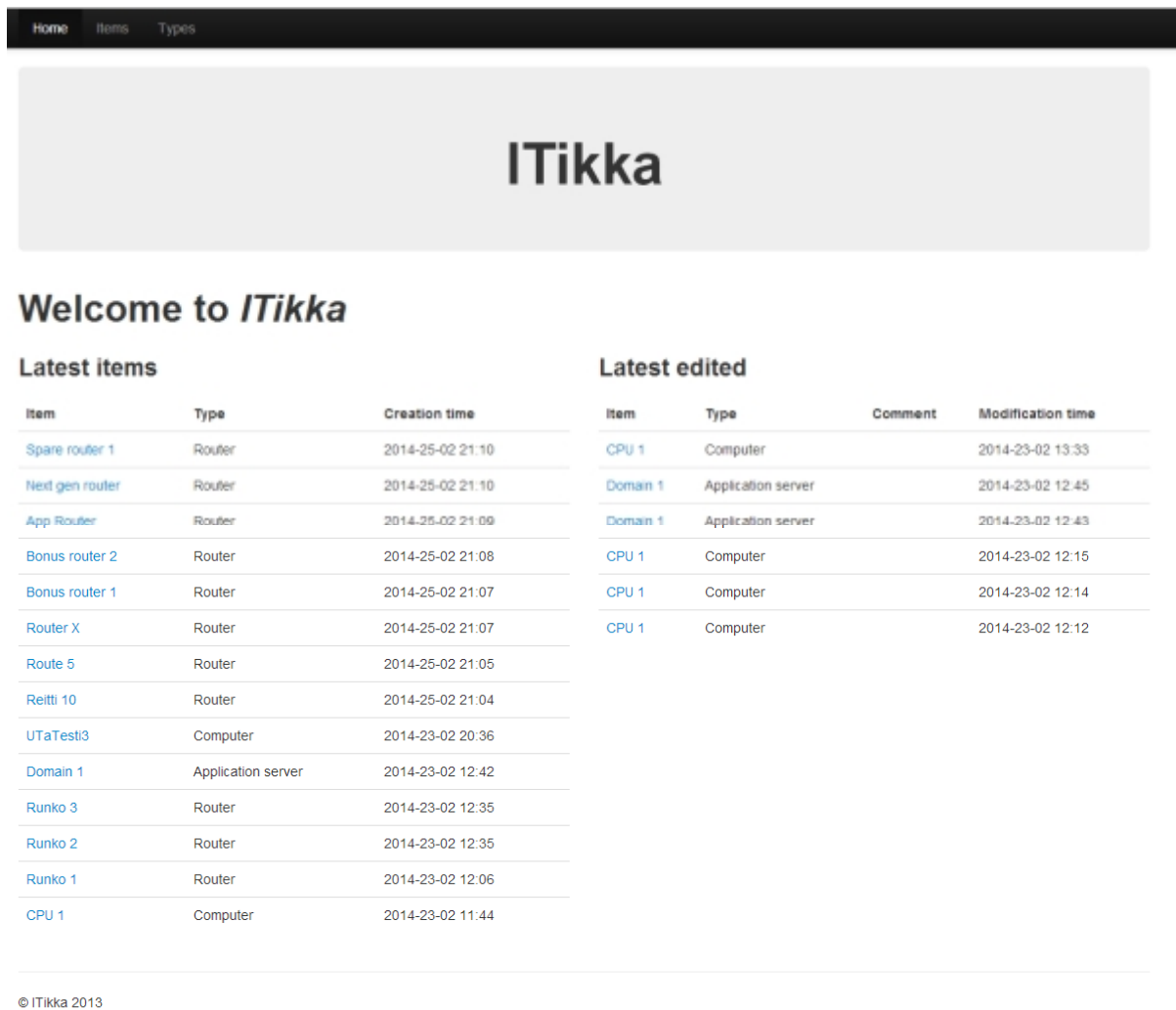
Edit relationships

Depends on	Dependency to	Id	Name	Type
------------	---------------	----	------	------

© ITikka 2013

Picture 1. Adding item.

The front page of the application lists the latest items added, and the last items modified.



Picture 2. Frontpage.

## Organization and management

The team consisted of 3 project managers and 3 team members. One developer quit the project. The team was divided upon preferences and skills to different types of tasks, with general responsibilities. The project managers divided responsibilities and tasks on a weekly basis.

1. Tuomo Hyttinen, Project Manager
2. Minna Mantila, Project Manager
3. Ville Murtonen, Project Manager
4. Sanna Rätty, UI / UX
5. Aarne Uotila, Coding / Database
6. Toni Vormisto, Database / Admin

Project was managed using Agile software project development methods. Meaning we had scrum type sprints of coding, with a session at the end of every sprint. Using the sessions, and redmine as ticketing system, we were able to set enough goals for the team to be able to function. The team was very self-managing so daily management was not needed, just an oversight on progress based on Redmine tickets.

## **Methods and tools**

Redmine – Useful for ticket and time tracking. Wiki functionality was too limited.

Flowdock – Useful for communication, not ready for replacing email due to visibility and reachability problems. I.e. People check email more often.

Modified Scrum – Good for this type of project that has people working on different hours and different periods. Provided the agility we needed to manage a software oriented team.

Yii – A solid framework, although a bit unpolished still. Provided the basic framework to which we built our software on top of. Easy to implement, quick to learn.

Postgresql – Common SQL database, easy to install, enough documentation for implementation.

Bootstrap – Good and easy to use/fast to learn framework for building a front-end for a web application. We used this to build a basic theme for our UI, and it worked really well for us.

Balsamiq – We did some UI sketches with this, seemed useful, but we didn't have that much use for it in this project.

Github – Version control in the cloud, pretty much as simple as it can get. Also has good documentation and newbie friendly training programs. We used it constantly for version control throughout the project.

## **Project phases and development model**

Our development model consisted of sprints, we had 7 sprints of about 3 weeks each.

Each sprint had a list of items for that sprint, with the possible backlog items from previous sprints. The tasks for each sprint were listed in redmine.

After each sprint, we had a face to face meeting, where we went through the issues on each sprint, possible problems and discussed about issues to be done in the next sprint. After this we created the decided tasks to redmine, if they were not already there.

We did not have any major phases in the project, they were all replaced with sprint. From the very beginning we had a quite clear idea about the implementation, and we could go from designing to prototyping in a very short time. Already in sprint 3 we had UI mockups and html-proto.

Other than that, we followed the course schedule for the documents and reports and customer meetings.



## Sprint 2

11/10/2013



9 issues (9 closed — 0 open)

### Related issues

~~Feature #1752~~: Basic overall layout, html+css  
~~Feature #1753~~: Add item, html+css  
~~Feature #1855~~: Create ActiveRecord models based on database schema  
~~Feature #1857~~: CRUD for types  
~~Feature #1858~~: CRUD for items  
~~Feature #1881~~: Create the actual UI for adding Items using html-proto

## Sprint 3

12/01/2013

Sprint 3: 11.11. - 1.12.2013



13 issues (13 closed — 0 open)

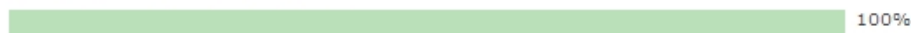
### Related issues

~~Feature #1856~~: UI-mockups, version 2  
~~Feature #1910~~: Add Template behaviour to Type  
~~Feature #1918~~: Add relationship  
~~Feature #1926~~: Implement adding Properties for Item  
~~Feature #1928~~: Type / template, html proto  
~~Feature #1929~~: html proto: Items version 2  
~~Feature #1945~~: Add Type

## Sprint 4

01/05/2014

Sprint 4: 2.12.2013 - 5.1.2014



8 issues (8 closed — 0 open)

### Related issues

~~Feature #2025~~: Remove Item  
~~Feature #2056~~: Implement example test case  
~~Feature #2061~~: UI design for dependency graphics

Picture 3. Sprints.

## Experiences

Our projects went really smoothly from the start; the only risk that realized for us was that one of the project members was not able to participate as much as he promised at the start. This only affected some of the “nice to have” features of the application, and didn’t have any real effect on being able to deliver the application to the customer.

Other than that, our team was balanced enough to provide enough coding and UI experience to manage the development process smoothly and the amount of project managers was more than enough to see through this sort of project.

The experiences we learned from this project are mostly related to the project management and the lack of oversight. There was no “main” project manager, so the management of the team and project was a bit complicated from time to time. Also the lack of steering group meant that the project managers didn’t really report to anyone, and thus there wasn’t that much for the project managers to do, while the team was doing their tasks. Also the customer coordination / steering was pretty loose, so the quality of the project deliverables is somewhat unknown.

For the future, for projects with multiple project managers, I would suggest pointing a lead project manager. Also each project should have proper steering group with understanding on how to steer a project, how to set and follow up on project progress and on how to manage the project manager(s).

## Statistics

Team size	Dev. model	Start date	End data		Hours
3+5	Modified Scrum	01.09.2013	07.03.2014		820

*Table 1: General project information.*

Acti- vity	Planni ng and manag ement	Req. specifi cation.	De- sign	Code	Integr ation and testing	Revie ws	Repair	Study	Other	Total
Hours	312.5	8	22	95.50	49	32	0	142	69.5	<b>730,5</b>
%	43	1,1	3	13,1	6,7	4,4	0	19,4	9,5	100%

*Table 2: Group effort by activity.*

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
20					

*Table 3: Requirements and high-level design outcomes.*

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
1	1	1	1	1	0

*Table 4: Design outcomes.*

Document	Pages	Versions
Preliminary analysis	8	FINAL
Project Plan	26	FINAL
Usability analysis	-	-
Requirements specification	1	FINAL
Design plan	-	-
User interface document	21	2.0
Test plan	14	FINAL
Test report	6	5.0
Usability test report	-	-
Final report	10	3.0
Project's story	7	1.0
Weekly reports	1	-

*Table 5: Documents.*

Language	PHP, HTML, Javascript
<b>Project name</b> itikka <b>Generated</b> 2014-03-04 20:15:39 (in 10 seconds) <b>Generator</b> <a href="#">GitStats</a> (version a266ddc), git version 1.7.0.4, gnuplot 4.2 patchlevel 6 <b>Report Period</b> 2013-10-06 15:11:05 to 2014-03-03 19:55:14 <b>Age</b> 149 days, 39 active days (26.17%) <b>Total Files</b> 2037 <b>Total Lines of Code</b> 668722 (671341 added, 2619 removed) <b>Total Commits</b> 102 (average 2.6 commits per active day, 0.7 per all days) <b>Authors</b> 5 (average 20.4 commits per author)	

*Table 6: Codelines.*

# Field Optimization Tool

## **Yleiskatsaus**

Field Optimization Tool -projektin ideana oli luoda Web-sovellus, jonka avulla voidaan sekä helpottaa että huomattavasti nopeuttaa asiakkaan tekemää työtehtävien allokointia. Web-sovelluksen tarkoituksena on korvata käytössä oleva vanha manuaalinen tapa jakaa työpäivän työt työntekijöille. Vanhan systeemin erityisenä heikkoutena on sen kyvyttömyys reagoida nopeasti tai älykkäästi poikkeustilanteissa, esimerkiksi työntekijän sairastuessa. Vanhassa systeemissä työtehtävien jakaminen hoidettiin käsin excel-tilukoita käyttäen ja tähän kului varsin paljon aikaa ja vaivaa. Myös tietojen säilyttäminen ja päivittäminen excel-muodossa on ollut vaivalloista.

Edellä mainitun päätavoitteen täyttämisen lisäksi toteutetun Web-sovelluksen avulla projektin asiakas pystyy myös muun muassa pitämään kirjaa työntekijöistään ja heidän tiedoistaan, omista asiakkaistaan sekä asiakkaiden kanssa solmituista sopimuksista. Web-sovelluksella asiakas voi suunnitella töiden jakamisen työntekijöille päiväkohtaisesti tulevaisuuteen ja lisäksi seurata reaaliajassa töiden valmistumista.

Pähkinänkuoressa uusi sovellus automatisoi aikaisemmin käsin tehdyn työn ja liittää sen osaksi pilvipalvelua. Pilvipalveluun siirryttäessä aikaisempi excel-tilukoiden pyörittely voidaan jättää pois ja käyttää helposti saatavilla olevaa web-palvelua töiden allokoinnin hoitamiseen automaattisesti. Pilvipalvelu myös tallentaa tiedot tietokantaan siten, että tiedot ovat saatavissa kirjautumisen kautta kunhan Internet-yhteys toimii.

## **Toteutettu sovellus**

Web-sovelluksen käyttäjät ovat asiakkaan henkilökuntaa. Käyttäjät on jaettu neljään eri luokkaan, jotka ovat (nousevassa tärkeysjärjestyksessä) työntekijä, myynti, manageri sekä admin. Osa sovelluksen toiminnallisuuksista on piilotettu ja estetty sellaisilta käyttäjiltä, joilla ei ole tarvetta tai oikeutta käyttää niitä.

Web-sovellusta käyttääkseen käyttäjän tulee siirtyä sovelluksen etusivulle, jossa käyttäjä suorittaa järjestelmään kirjautumisen omalla käyttäjätunnuksellaan ja salasanallaan. Kirjaututtuaan sisään käyttäjälle avataan jokin ennalta määrätty näkymä Web-sovelluksesta, riippuen siitä, minkä tason käyttäjä on kyseessä.

Toteutettu Web-sovellus jakautuu viiteen erilaiseen näkymään. Nämä näkymät ovat työt-näkymä, asiakas-näkymä, työntekijät-näkymä, optimointi-näkymä sekä omat tiedot-näkymä. Nämä näkymät ovat yksi esimerkki siitä, miten toiminnallisuuksia eritasoisten käyttäjien välillä on rajattu. Esimerkiksi työntekijä tason käyttäjä voi käyttää ainoastaan työt- ja omat tiedot-näkymiä, kun taas manageri tason käyttäjä voi käyttää kaikkia näkymiä.

## Työt-näkymä

Työt-näkymä on tarkoitettu ensisijaisesti työntekijöille. Työt-näkymän toimii sekä tableteilla, että tavallisilla tietokoneilla. Työt-näkymässä työntekijä näkee päiväkohtaisesti tekemättömät ja tehdyt työt ja lisäksi jokaisen työn työohjeen. Työntekijä voi myös lisätä kommentin työlle. Työt-näkymässä on myös kartta, jonka avulla työntekijän voi katsoa kartalta seuraavan työkohteen sijainnin. Kartalla näytetään myös reittiehdotus edellisen ja valitun työkohteen välille.

Field Optimization tool


Työt


Kalenteri

Omat tiedot

Hei, Jore!





Kirjaudu ulos





25.02.2014

Työpari: Lissu Liusukka

Tehty	Asiakas	Aika	Kesto	Osoite	Työohje	Kommentti
	Tampereen yliopisto	08:00 - 14:00	1 h	Kalevantie 3		
	TAMK	08:00 - 14:00	1 h	Teiskontie 33		

Edellisten toimintojen lisäksi työt-näkymän tiedot voidaan esittää myös helposti tulostettavassa muodossa. Tämän toiminnallisuuden avulla työntekijät saavat halutessaan tulostaapäivän työt paperille. Lisäksi manageri tason käyttäjät voivat tulostaa valitsemansa päivän kaikkien työntekijöiden työt yhdessä raportissa.

## Asiakas-näkymä

Asiakas-näkymästä voidaan hallinnoida asiakastietoja. Näkymästä voidaan syöttää järjestelmään uusia asiakkaita, muokata olemassa olevien asiakkaiden tietoja ja myös poistaa asiakastietoja. Asiakkaasta tallennettavia tietoja ovat asiakkaan nimi, lähiosoite, puhelinnumero, kaupunki, sähköposti ja postinumero. Tämän lisäksi näkymästä voidaan hallita asiakkaan kanssa solmittuja sopimuksia. Sopimuksista tallennetaan sopimuksen alkamispäivä, päättymispäivä, työajankohta, työn kesto, toistuvuus, työpäivät ja tiedot niistä työntekijöistä, joille työ ensisijaisesti allokoidaan ja myös niistä, joille työtä ei saa allokoida. Sopimukselle voidaan tallentaa myös pidempi työohje. Asiakkaiden tietoja voidaan viedä sisään ja ottaa ulos myös csv-tiedostoina.

## Asiakkaat

yliopisto

Tampereen yliopisto

Uusi asiakas Vie/Tuo +

## Tampereen yliopisto

Tiedot Sopimukset Kalenteri

### Helmikuu 2014

kuukausi viikko päivä tänään < >

V	ma	ti	ke	to	pe	la	su
5	27	28	29	30	31	1	2
6	3	4	5	6	7	8	9
7	10	11	12	13	14	15	16
8	17	18 -, 1h, ko 08:00-14:00	19	20 -, 1h, ko 08:00-14:00	21	22	23 Pena Makkonen, 1h, ko 08:00
9	24 Jore Aaltonen & Lasku Laskua, 1h, ko 08:00	25	26	27 -, 1h, ko 08:00-14:00	28	1	2
10	3 -, 1h, ko 08:00-14:00	4	5	6 -, 1h, ko 08:00-14:00	7	8	9

## Työntekijät-näkymä

Työntekijät-näkymässä voidaan hallita yrityksen työntekijöitä ja samalla web-palvelun käyttäjiä. Työntekijöitä voidaan lisätä, muokata ja poistaa tämän näkymän kautta. Työntekijästä tallennettavia tietoja ovat etunimi, sukunimi, käyttäjänimi, sähköposti, osoite, postinumero, postitoimipaikka, puhelinnumero. Perustietojen lisäksi järjestelmään on mahdollista tallentaa työntekijän työauton rekisterinumero, mahdolliset pätevyyskortit sekä lomat. Lisäksi järjestelmässä on vapaamuotoinen tekstikenttä työntekijän omille kommentteille sekä oma tekstikenttä työnantajan huomioille, jonka sisältöä työntekijä ei itse näe. Järjestelmään tallennetaan myös salasana sekä käyttäjärooli työntekijälle. Kuten asiakkaiden puolella, myös työntekijöiden tietoja voidaan viedä sisään ja ottaa ulos csv-tiedostoina.

Field Optimization Tool Työt Asiakkaat Työntekijät Optimointi Omat tiedot Hei, Jarkko! Kirjaudu ulos

Henkilöstö

Jore Aaltonen

Kaikki Työntekijät Poistettut

Jore

Jore Aaltonen

Tiedot Kalenteri

Etunimi Jore Sukunimi Aaltonen Käyttäjänimi jore

Sähköposti Sähköposti Osoite Hämeenkatu 1 Postinumero 33100

Postitoimipaikka Tampere Puhelinnumero 112 Auton rekisterinumero rekisterinumero

☒ Ajokortti ☐ Työturvallisuuskortti ☐ Ensiapu ☐ Henkilönostin  
☒ Aamu ☐ Iltä ☐ Viikonloppu  
Lomat  Rooli työntekijä  

Lisää loma Poista loma

Muuta

Mukava tyyppi

Työnantajan huomioita

Muuta

Lisää henkilö Vie tiedot Tuo tiedot

Tallenna Poista Vaihda salasana

## Optimointi-näkymä

Optimointi-näkymässä manageri tason käyttäjät voivat tehdä päivän työtehtävien jakamisen työntekijöille. Optimointi-näkymässä on nimensä mukaisesti myös optimointi toiminto. Optimoinnin tehtävänä on suorittaa työtehtävien jako mahdollisimman hyvin tiettyjen ehtojen puitteissa.

Optimointi-näkymästä nähdään valitun päivän työtehtävät vasemmassa sarakkeessa. Työpaikan ja työntekijän sijainnit näytetään oikeanpuoleisella kartalla. Työtehtävistä voidaan näyttää valittavaksi kaikki, aamuvuorossa olevat, iltavuorossa olevat tai ei mitään työpareja. Valitut tehtävät näytetään oikeanpuoleisella kartalla.

Toiminnot-valikosta voidaan valita työtehtävien optimointiin ja allokontiin liittyviä toimintoja. Tämän valikon alta löytyy ”muokkaa päivän työntekijöitä”-toiminto, jonka avulla voidaan lisätä ja poistaa työntekijöitä päivälle. Toinen toiminto on ”poista työt tekijöiltä”, joka asettaa kaikki päivän työt asettamattomiksi. Kolmas toiminto, ”optimoi uudelleen”, asettaa automaattisesti työt työpareille optimointialgoritmin avulla.

Karttanäkymän lisäksi löytyy myös taulukkonäkymä, jossa työtehtävien jakautumista ja työpareja voidaan muokata Drag and Droppaamalla.

10





## Menetelmät ja välineet

Projektinhallintatyökalut:

- Google Drive
- Redmine
- Balsamiq

Kaikki projektinhallintatyökalut toimivat todella hyvin projektimme käyttötarkoituksiin. Erityisen hyödyllinen oli Balsamiq, jonka avulla pystyi todella helposti ja pienellä vaivalla suunnittelemaan käyttöliittymiä.

Kehitystyökalut:

- Eclipse
- Visual Studio
- TortoiseSVN

Kehitystyökalut olivat kaikille projektilaisille tuttuja ja ne sopivat todella hyvin PHP, HTML ja JavaScriptin tekemiseen. TortoiseSVN:n avulla versionhallinnan käyttö oli helppoa.

Ohjelmistokehykset:

- AngularJS
- jQuery
- Less Css

Selain puolella käytimme eri ohjelmistokehyksiä. AngularJS:n ja jQuery:n avulla saimme käyttäjälle paremman käyttökokemuksen. Less Css:ää käyttäen saimme tehtyä helposti esteettisesti miellyttävän käyttöliittymän. Ohjelmistokehyksien käyttämisen opetteluun meni hieman aikaa, mutta loppujen lopuksi näiden käyttö kannatti. Ohjelmistokehyksien takia kehittäjien ei tarvinnut tehdä niin paljon koodia verrattuna siihen, että he olisivat tehneet ilman ohjelmistokehyksiä.

Kommunikointityökalut:

- Skype
- Sähköposti

Lähes kaikki projektiin liittyvä keskustelu käytiin Skypen välityksellä ryhmäkeskustelussa. Skype toimi hyvin ryhmän sisäiseen kommunikointiin todella hyvin. Skype keskusteluista jää myös historia, joka osoittautui monesti erittäin hyödylliseksi. Sähköpostia käytimme asiakkaan ja kurssin ohjaajien kanssa keskusteluun sekä viikkoraporttien jakeluun.

## Projektin vaiheet ja kehitysmalli

Käytimme kehitysmallina muunneltua Scrum kehitysmallia. Pidimme kahden viikon mittaisia sprinttejä normaalin yhden viikon sijaan. Jokaisen sprintin jälkeen pidimme ryhmän sisäisen katselmoinnin, jossa tarkastelimme mitä ryhmäläiset saivat aikaiseksi Sprintin aikana. Pidimme joka viikko viikkotapaamisen, joissa käytiin läpi mitä ryhmäläiset ovat tehneet kuluneen viikon aikana ja mitä he tulevat tekemään seuraavalla viikolla. Viikkopalaverissa oli myös mahdollisuus keskustella muista projektiin liittyvistä asioista.

<b>Sprint</b>	<b>Tapahtuma</b>	<b>Päiväys</b>
Sprint 0		04.09.2013 - 30.09.2013
	Lyhyt projektiesitys	18.9.2013
	Ennakko analyysi	23.9.2013
Spring I		30.09.2013 - 21.10.2013
	Projektisuunnitelma	25.9.2013
	Projektisuunnitelman katselmus	7.10.2013
	Sprint I Katselmus	21.10.2013
Spring II		21.10.2013 - 4.11.2013
	Personal report I	1.11.2013 - 7.11.2013
	Sprint II Katselmus	4.11.2013
Spring III		4.11.2013 - 18.11.2013
	Sprint III Katselmus	18.11.2013
Spring IV		18.11.2013 - 2.12.2013
	Sprint IV Katselmus	2.12.2013
Spring V		2.12.2013 - 16.12.2013
	Kurssi katselmus 1	2.12.2013
	Kurssi katselmus 2	16.12.2013
	Sprint V Katselmus	16.12.2013
Spring VI		16.12.2013 - 13.1.2014

	Personal report II	1.1.2014 - 16.1.2014
	Sprint VI Katselmus	13.1.2014
Spring VII		13.1.2014 - 27.1.2014
	Sprint VII Katselmus	27.1.2014
Spring VIII		27.1.2014 - 10.2.2014
	Kurssi katselmus 3	3.2.2014
	Sprint VIII Katselmus	10.2.2014
Spring IX		10.2.2014 - 24.2.2014
	Sprint IX Katselmus	24.2.2014
Projektin lopetus		24.2.2014 - 3.3.2014
	Testaus raportti	
	Loppuraportti	
	Projektitarina	
	Viimeinen tapaaminen	3.3.2014
	Loppuesitys	5.3.2014
	Personal report III	5.3.2014

Taulukko 1: Aikataulu

## Kokemukset

Kaikki projektiin osallistuneet saivat projektista käytännön kokemusta osallistumisesta oikeaan ohjelmistokehitysprojektiin. Projektissa pidimme erityisen hyödyllisenä sitä, että projektissa oli mukana oikea yliopiston ulkopuolinen asiakas ja pääsimme pureutumaan oikean yrityksen töiden allokointiongelmiin. Projektin onnistui odotusten mukaisesti. Toimitimme asiakkaalle toimivan töiden optimointityökalun.

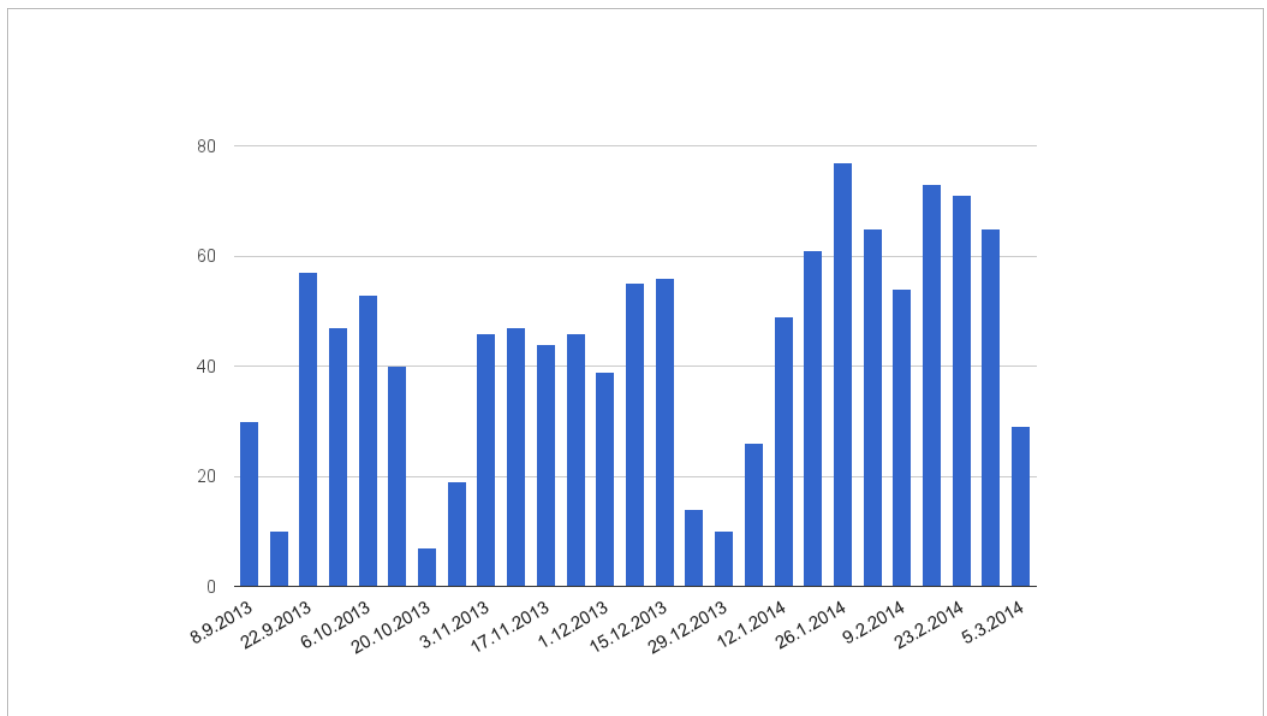
Projektin alkuvaiheessa saimme asiakkaalta suhteellisen vapaat kädet ohjelman toteuttamiseen. Tämä toi projektin edetessä mukanaan omat haasteensa, sillä kehitystiimillä oli eri näkemys joidenkin toiminnallisuuksien osalta kuin asiakkaalla. Projektin aikana pyrimme vastaamaan mahdollisimman hyvin asiakkaalta tulleisiin uusiin ideoihin. Vaatimusten muuttumiseen sopeutuminen oli mahdollista Scrumin joustavuuden ansiosta. Projektin loppuvaiheessa löysimme asiakkaan kanssa yhteisen näkemyksen toteutettavista ominaisuuksista. Tämä onnistui karsimalla alkuperäistä ideaa hieman, jotta saimme aikaiseksi toimivan ohjelman.

Suurempia riskejä emme projektin aikana kohdanneet. Kaikki projektiryhmäläiset pysyivät mukana koko projektin ajan ja saivat toteutettua annetut tehtävät. Suurimmat haasteet toivat projektiryhmäläisten vaihtelevat aikataulut, sillä jokainen ryhmäläinen opiskeli samaan aikaan muita kursseja tai kävi töissä. Myös uudet vaatimukset projektin edetessä toivat lisähaasteita. Selviydyimme kuitenkin näistä haasteista hyvin ja projekti saatiin päätökseen aikataulussa.

## Tilasto

Ryhmän koko	Kehitysmalli	Aloituspäivä	Lopetuspäivä	Päiviä	Tunnit
6	Scrum	4.9.2014	5.3.2014	182	1200

Taulukko 2: Yleistä



Kaavio 1: Työtunnit viikoittain

Aktiivisuus	Suunnittelu ja hallinto	Vaatimukset	Suunnittelu	Ohjelmointi	Integrointi ja testaaminen	Katselmointi	Korjaaminen	Opiskelu	Muuta	Yhteensä
Hours	370	0	13	464	22	75	53	155	38	<b>1190</b>
%	31	0	1,1	39	1,8	6,3	4,4	13	3,2	100%

Taulukko 3: Aktiivisuus

Vaatimukset	Käyttöliittymät	Tietokantakaaviot	Tietokantataulut
56	6	1	14

Taulukko 4: Vaatimukset ja suunnittelu dokumentit

Dokumentit	Sivut	Versiot
Ennakoanalyysi	5	1
Projektisuunnitelma	15	3
Testiraportti	9	3
Loppuraportti	13	3
Projektitarina	10	1
Käyttöohjeet, työntekijä	4	1
Käyttöohjeet, manageri	18	1
Viikkoraportit	1	1

Taulukko 5: Dokumentit

Ohjelmointikieli	PHP	Java Script
LOC	3205	2754
Luokat	19	-
Funktiot	160	364
Versiohallinnan versio	446	446

Taulukko 6: Koodirivit

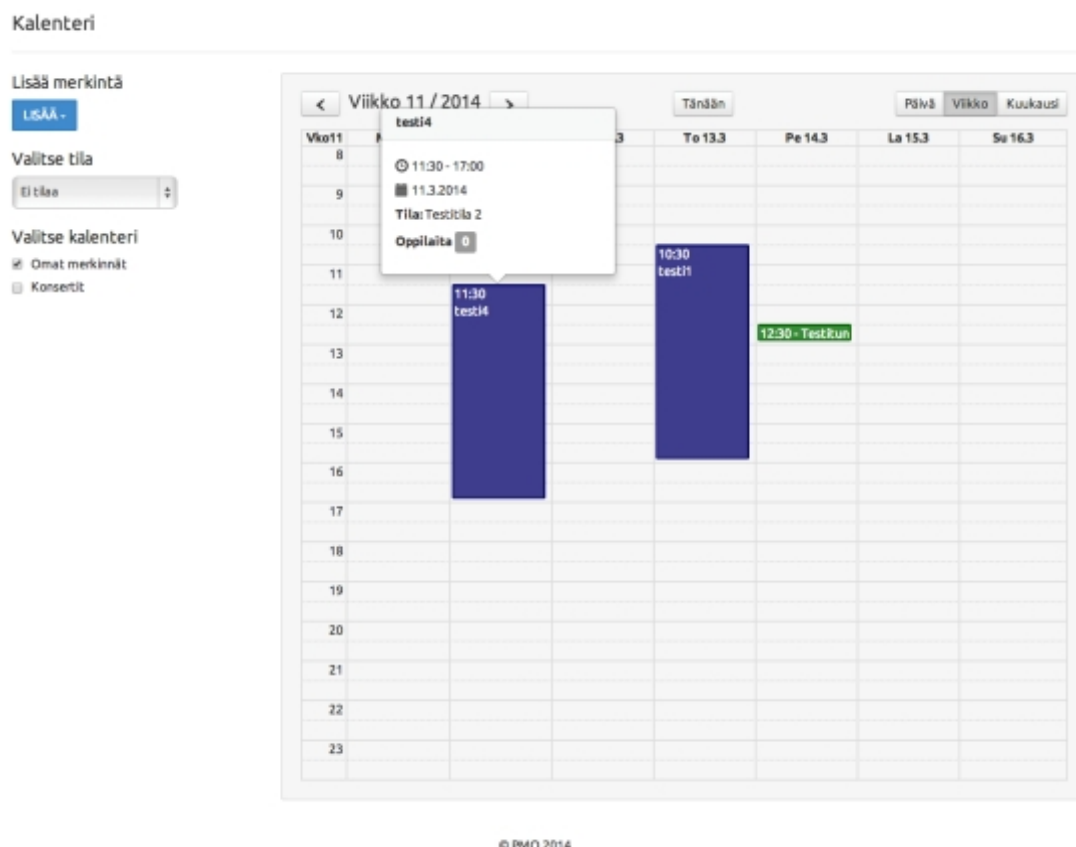
# Pirkanmaan musiikkiopisto

## Yleiskuvaus

Projektissa toteutimme Pirkanmaan musiikkiopiston käyttöön responsiivinen verkkosivuston, jolla voidaan hallita musiikkiopiston tilavaraukset sekä lisätä näihin varauksiin opettajia ja oppilaita. Tämän lisäksi järjestelmän tulee olla laajennettavissa kattamaan seuraavat asiat:

- Hallinnoida uudistuva materiaalipankkia
- Tuottaa opettajakohtaiset lukujärjestykset ja raportit
- Oppilaiden lukukausi-ilmoittautumiset.

Vaikka järjestelmä toteutetaankin Pirkanmaan musiikkiopistolle, tarkoituksena oli tuottaa yleiskäyttöinen ohjelmisto, jota voidaan hyödyntää muissakin ympäristöissä.



Kuva 1. Kalenterin perusnäkö

## Organisointi ja hallinta

Projektiryhmä koostui kolmesta projektipäälliköstä sekä kolmesta projektityöntekijästä.

Päälliköinä toimivat:

- Markus Salomaa
- Ville Siltala
- Jukka Suorsa

Projektityöntekijöinä olivat:

- Matti Palonen
- Aleksi Wikman
- Veera Kuusikko

Projektipäälliköiden työt jakaantuivat siten, että Markus hoiti dokumentointia ja projektin yleisten asioiden hoitamista. Ville vastasi JavaScript-ohjelmoinnista, sovelluksen yleisestä arkkitehtuurista, katselmointien pitämisestä sekä projektiryhmän opastamisesta. Jukan tehtäviä olivat yksikkötestauksen toteuttaminen, koodin tuottaminen ja dokumentointi.

## Menetelmät ja työkalut

Ohjelmointikielenä olivat Python ja JavaScript, kirjastoista käytössä olivat Django, AngularJS ja Twitter Bootstrap. Sivujen ulkoasua muokattiin CSS:llä.

Projektiin liittyviä dokumentteja ja ohjeita keräsimme Redmineen ja Flowdockiin, joista ne ovat kaikkien saatavilla koko ajan. Viikkoraportointien julkaisu suoritettiin sähköpostin välityksellä. Raportit välitettiin jokaisen viikon maanantaina projektiryhmälle, kurssin vetäjille Timo Poraselle ja Pekka Mäkihaholle sekä asiakkaan yhteyshenkilöille. Projektin yleinen dokumentointi suoritettiin Google Drivellä. Tuntikirjanpitoa ja tehtävälistausta ylläpidettiin Redminessä.

Projektiryhmän välinen kommunikointi hoitui Flowdock-keskustelukanavan kautta. Yleisen keskustelun lisäksi tämän kanavan kautta hoidettiin toisten ryhmäläisten auttaminen projektiin liittyvissä asioissa, viikkopalavereiden ja työpajojen sopiminen sekä asiakaskommunikointi.

## Projektin vaiheet ja kehitysmalli

Projektissa käytimme Scrum-kehitysmallia, jota kuitenkin muokkasimme yliopiston kurssille paremmin soveltuvaksi. Emme pitäneet päivittäisiä Scrum-tapaamisia, vaan kokoonnuimme viikkopalaveriin, jotka projektin alussa pidimme lähes poikkeuksetta keskiviikkoisin. Projektin edettyä vaihtelimme päivää sen mukaan, mikä sopisi kaikille parhaiten. Myöhemmin kurssin edetessä projektitapaamiset pidettiin workshop-tyyppisinä. Koko projektin ajan palaverit ja kokoontumiset toimivat moitteettomasti ja koko ryhmä saatiin kasaan lähes joka kerta.

Työnjako suoritettiin projektin alusta lähtien melko järjestelmällisesti, sillä tietty henkilö vastasi omasta projektin osa-alueestaan. Tämä osoittautui ainoaksi järkeväksi vaihtoehdoksi, koska opeteltavia uusia tekniikoita oli paljon, eikä kellään ollut mahdollista opetella kaikkia työkaluja itse. Jokainen oli valmis auttamaan muita omien töidensä ohella.

Pvm	Tapahtuma
26.9.	Esitutkimuksen katselmointi
7.10.	Projektisuunnitelman katselmointi
23.10- 20.11	1. sprint
1.-7.11.	Personal report
20.11	1. katselmointi
20.11 - 29.1	2. sprint
29.1	2. katselmointi
29.1 - 4.3	3. sprint
1.-14.1.	Personal report
4.3	3. katselmointi
maalisk.	Tuotteen viilausta ja testausta
31.3	Tuotteen luovutus
31.3.	Loppuraportti
huhtik.	Personal report



## Kokemuksia

Saimme projektin hyvin alkuun syksyllä 2013, ja vaatimusmäärittelyt selkeytyivät kohtalaisen nopeasti ensimmäisen kuukauden aikana. Ryhmätyöskentely ja palaveriin osallistuminen on ollut koko kurssin ajan kiitettävällä tasolla, joten sen puolesta projektityöskentely on ollut helppoa.

Suurimmat ongelmat kurssin aikana kohdistuivat valittuihin teknologioihin. AngularJS ja Django olivat ryhmälle uusia teknologioita, jonka seurauksena uuden opiskeltavaa oli paljon ja ohjelmistokehitys hidasta. Jälkikäteen ajateltuna meidän olisi pitänyt rohkeammin vaihtaa käytettäviä tekniikoita, kun havaitsimme kuinka työlästä Angularin ja Django-yhteensovittaminen on. Yhtenä vaihtoehtona olisi ollut pelkästään Python-sovelluksen tekeminen, jolloin myös uuden opeteltavan määrä olisi puolittunut, ja useampi henkilö olisi saanut paremman kokonaiskuvan toteutuksesta.

Software Project Management- sekä project work -kurssit koettiin erittäin haasteellisina, mutta tarpeellisina. Kurssit opettivat jokaiselle ryhmäläiselle paljon uusia asioita, joita voimme hyödyntää tulevaisuuden opinnoissa sekä työelämässä. Teknologioiden kanssa tapahtuneet virhevalinnat koettiin loppuen lopuksi opettavana kokemuksena, josta ryhmäläiset saivat hyviä ensikosketuksia tulevaisuudessa käytettäviin teknologioihin.

Kurssilla toteutunut projekti oli monelle ryhmäläiselle ensimmäinen hieman pidempi kestoinen projekti, joten sen aikana on saatu paljon merkittävää tietoa ohjelmistoprojektin läpiviemisestä, vaatimusmäärittelyistä, työmääräarvioista ja ryhmäyhteistyöstä. Jokainen on päässyt toteuttamaan itseään ryhmässä ja saanut tehdä töitä sellaisten asioiden parissa, joita ei aiemmin ole tehnyt.

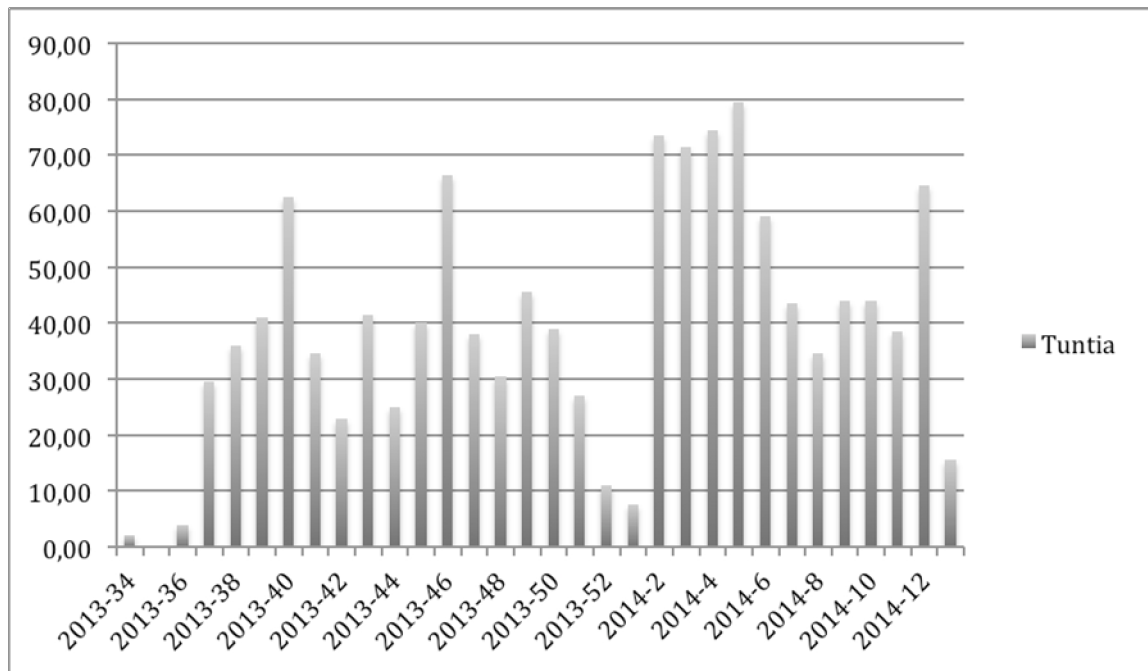
## Tilastoja

Ryhmän koko	Dev. model	Start date	End data	Days	Hours
3+3	Scrum	11.9.2013	31.3.2014	202	1246.50

*Taulukko 1: Yleistä tietoa projektista*

Aktivi teetti	Plann ing and mana geme nt	Req. speci ficati on.	De- sign	Code	Integ ration and testin g	Study	Other	Total
Tunnit	255.5	21.5	19	601	78.5	199	72	<b>1246.50</b>
%	20.50	1.72	1.52	48.22	6.30	15,96	5.78	100%

*Taulukko 2: Projektiryhmän kokonaistunnit.*



Taulukko 3: Työtuntien jakaantuminen

Ehdotettu	Hyväksytty	Toteutettu	Vahvistettu	Hylätty
0	24	15	0	9

Taulukko 4: Vaatimukset

Dokumentti	Sivumäärä	Versio
Esitutkimus	7	1
Projektisuunnitelma	23	1
Loppuraportti	25	1
Projektin tarina	5	1
Presentaatiot	3	1
Palaverimuistiinpanot	17	1
Viikkoraportit	27	27
Testidokumentaatio	2	1

Taulukko 5: Dokumentit.

# Math.fi



## Overview

Math.fi is an open mathematical learning environment in internet for 13 -16 year old pupils (upper comprehensive school, grades 7-9).

The purpose of Math.fi-system is to motivate pupils to learn mathematics independently and give them realistic response. Pupils are encouraged to be responsible for their learning.

Project objectives:

- development of usability
- development of reportage
- development of response properties
- development of pupils guidance and independent learning.

## Organization and management

### *Managers:*

Sami Pulli	scrum master
Pasi Riihiviita	project management
Jari Smedberg	usability design, testing, communication with the client

### *Project team:*

Senja Ampuja	usability design, testing
Antti Eikonsalo	coding, database management
Ville Kuparinen	usability design, testing
Samuel Lehtonen	coding

The project team had weekly scrum meetings, sprint meetings, and used Facebook for instant communication. Google Drive and Google Calendar were used to organize meetings and data. The Bug Genie was used for requirements engineering and managing issues and tasks. Google Drive was used to manage working hours.

## Methods and tools

- Programming language: PHP 5.4, JavaScript
- Test tools: Jenkins/Sonar
- Programming environment: Netbeans
- Framework: CakePHP 2.0
- Database: MySQL
- Version management: Subversion (svn)
- Project management, requirements process: The BugGenie (TBG)
- Communication: Google Gmail/Drive/Calendar

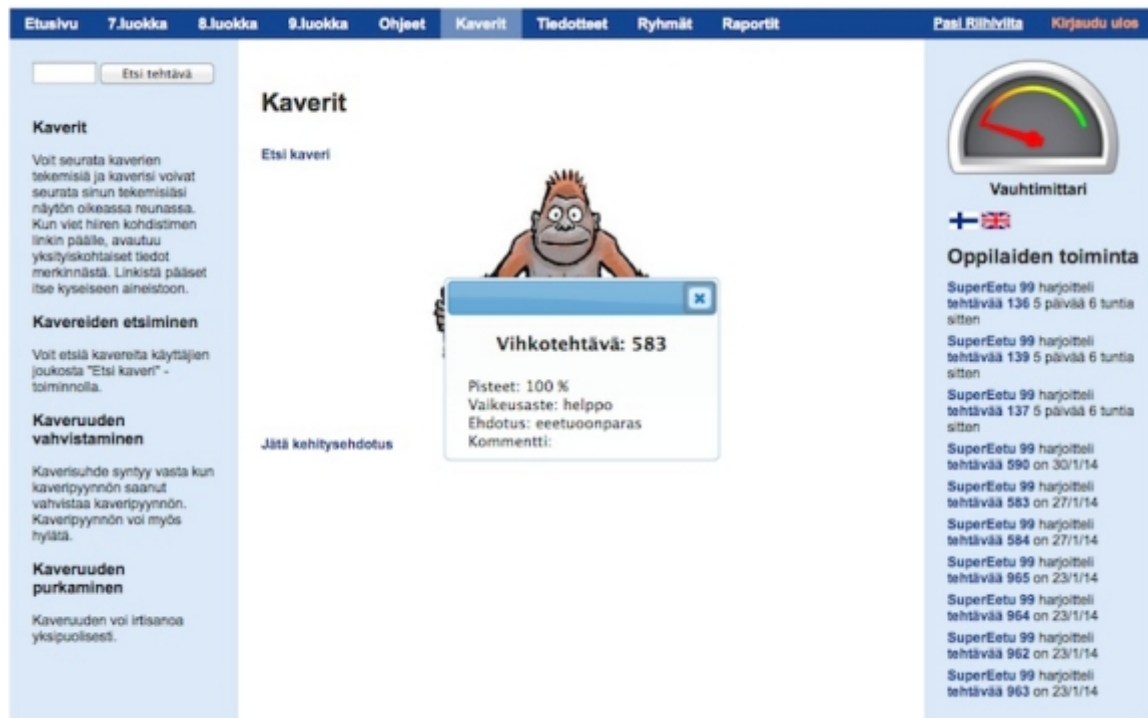
## Project phases and development model

Our development model was modified scrum with 4 week sprints. Sprint 4 was 5 weeks and sprint 5 2 weeks. We held weekly meetings most of the time and organized some workshops too. We used Facebook and e-mail to communicate between the weekly meetings.

Project started 18.09.2013  
Preliminary analysis meeting 27.09.2013  
Project plan inspection 11.10.2013  
First sprint 10.10.2013 - 14.11.2013  
First review 04.12.2013  
Second sprint 14.11.2013 – 04.12.2013  
Second review 09.01.2014  
Third sprint 04.12.2103 – 09.01.2014  
Third review 13.02.2014  
Fourth sprint 09.01.2014 – 13.02.2014

Fifth sprint 13.02.2014 – 27.02.2014  
Final meeting and project ending 12.03.2014

Kuva uudesta projektin tuottamasta ominaisuudesta. Feature request #12 - Reaaliaikainen seuranta: viemällä hiiren osoittimen merkinnän päälle avautuu popup, jossa näkyy oppilaan päiväkirjamerkinnät ko. tehtävästä.



## Experiences

### *Lack of proper requirement management process*

The project did not have a proper requirement management process during the first three to four months. The process was undefined and developers did not have clear objectives for sprints. At the beginning of the project the customer did not provide us with detailed goals and issues. This caused difficulties in managing the project efficiently and selecting issues to sprints. The team was not able to foresee coming efforts and was unaware of the state of the whole project. During the Christmas period one of the managers created in co-operation with the client a requirement process which was adopted during the last two sprints of the project.

### *Communication problems and inactivity*

The project had some problems with communication and inactivity among the team and managers. Although the responsibilities for team members and project managers were defined at the beginning of the project there was some misunderstandings and unawareness about the project goals and teamwork. The team did not communicate enough with each other although they were encouraged to plan the sprints together. Project managers committed similar mistakes though. It became obvious that Scrum is not the best way to handle these kind of projects where students have other tasks or work to do. Living in another town does not simplify the challenges because team

meetings and workshops have to be agreed well beforehand and there is always someone who is not able to participate no matter what. Typical reasons for inactivity were other trips, work or school getting in the way. Most of the trips, work and other delays were informed in advance though.

### ***Problems with technology***

The team had some problems with Jenkins / Sonar environment and we were unable to do unit tests during the autumn semester and getting help from the university was difficult.

### ***Testing process not designed***

Testing plan should be written in the beginning of the project. Testing was not properly applied during the sprints and had to be done at the end of each sprint. This caused timing problems and all issues included in the sprint were not ready and tested when they should be presented to the client. Testing is an essential part of programming.

## **Statistics**

Team size	Dev. model	Start date	End data	Days	Hours
3+4	Modified Scrum	18.9.2014	12.3.2014	176	1250

*Table 1: General project information.*

Acti- vity	Planni ng and manag ement	Req. specifi cation.	De- sign	Code	Integr ation and testing	Revie ws	Repair	Study	Other	Total
Hours	472	62	87	142	139	79	13	195	61	<b>1250</b>
%	38	5	7	12	11	6	1	16	4	100%

*Table 2: Group effort by activity.*

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
65	Maintained in TheBugGeni e	2		1	1

*Table 3: Requirements and high-level design outcomes.*

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
		1	13		

*Table 4: Design outcomes.*

Document	Pages	Versions
Preliminary analysis	7	1
Project Plan	19	5
Usability analysis	7	1
Requirements specification	Maintained in TheBugGenie	
Design plan	10	1
User interface document	23	1
Test plan	7	1
Test report	9	1
Usability test report	21	1
Final report	14	1
Project's story	5	1
Weekly reports	23	23

*Table 5: Documents.*

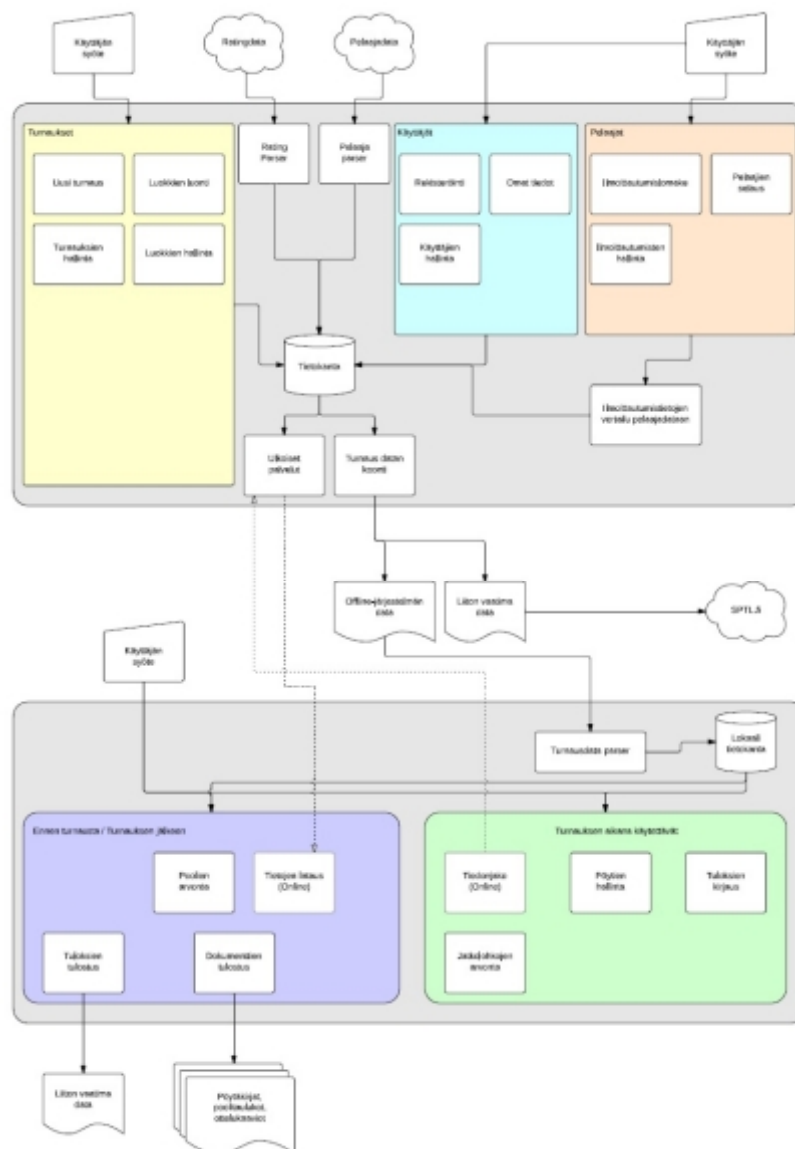
Language	PHP, Javascript
LOC	6375
SLOC	4395
Reused code	
Reused and modified	
Classes	56
Functions	197
Code revisions	Build 189, revision 1132

*Table 6: Codelines.*

# Pingismaisteri

## Yleiskatsaus

Pingismaisterin tarkoitus on helpottaa pöytätennisturnausten järjestämistä, turnausten ottelutulosten kirjaamista sekä pelaajatietojen hallinnointia. Järjestelmä on Web-pohjainen sovellus, jota pystyy käyttämään internetyhteyden kanssa sekä tarpeen vaatiessa myös ilman internetyhteyttä. Ilman internetyhteyttä käyttäminen on tarkoitettu lähinnä turnausten aikaisten toimintojen suorittamiseen, kuten turnauspoolien arvontaan ja ottelutulosten kirjaamiseen. Ilman internetyhteyttä käyttäminen vaatii kuitenkin esimerkiksi XAMPP-ohjelmiston. Järjestelmän yleiskuvaus on mallinnettu kuvassa 1.



Kuva 1



## Toiminnallisuus

Tässä luvussa käydään läpi sovelluksen keskeisimmät ominaisuudet. Keskeisimpiä ominaisuuksia ovat käyttäjien sekä pelaajien lisääminen ja hallinnointi, seurojen hallinnointi, turnauksien lisääminen ja hallinnointi, turnauksiin ilmoittautuminen, turnauspoolien arvonta ja hallinnointi sekä turnauksen aikaisten tietojen kirjaaminen.

Pingismaisterijärjestelmän käyttö vaatii käyttäjien rekisteröitymisen. Rekisteröityneet käyttäjät pystyvät kirjautumaan omilla tunnuksillaan sisälle järjestelmään. Järjestelmään rekisteröityminen on esitetty Kuvassa 2.

### PINGISMAISTERI

[Kirjaudu sisään](#) [Rekisteröidy](#)

**Nimi**

**Sähköpostiosoite**

**Puhelinnumero**

**Salasana**

**Salasana uudestaan**

[Rekisteröidy](#)

*Kuva 2*

Järjestelmä jakautuu normaalin käyttäjän näkökulmasta neljään osa-alueeseen: Turnausten hallinnointiin, turnausluokkien hallinnointiin seurojen hallinnointiin ja omien tietojen hallinnointiin. Turnausten osalta pystytään luomaan uusia turnauksia ja muokkaamaan sekä tarkastelemaan olemassa olevia. Turnauksen hallinnointi on esitetty Kuvissa 3, 4, 5 ja 6.

### PINGISMAISTERI

[Turnaukset](#) [Turnausluokat](#) [Seurat](#) [Oma tili](#) [Kirjaudu ulos](#)

#### Turnaukset

#	Nimi	Luokkia	Toiminnot
1	PT 75:n kansalliset 22.02.2014 - 22.02.2014, Hatanpään Koulu	14	<a href="#">Ilmoittaudu</a> <a href="#">Lisätietoja</a> <a href="#">Muokkaa tietoja</a> <a href="#">Poista</a>

[Luo uusi turnaus](#)

*Kuva 3: Turnauksen päänäkymä*

## PT 75:n kansalliset

Turnauksen tiedot	Ilmoittautuneet
<b>Turnauksen nimi</b> PT 75:n kansalliset <b>Järjestäjä</b> PT 75 Ylituomari: Jukka Nieminen	<b>22.02.2014</b>
<b>Ajankohta</b> 22.02.2014 - 22.02.2014 <b>Sijainti</b> Hatanpään Koulu	<b>MK</b> <ul style="list-style-type: none"><li>Syrjänen, Tapio (PT 75)</li><li>Tuomola, Mika (PT 75)</li><li>Jokinen, Antti (PT 75)</li><li>Tran, Huy (PT 75)</li></ul>
<b>Leikkuripäivä</b> 20.02.2014	<b>M-2000</b> <ul style="list-style-type: none"><li>Lallo, Ismo (TuKa)</li><li>Syrjänen, Tapio (PT 75)</li><li>Kurunmäki, Kyösti (Wega)</li><li>Faily, Jusef (PT 75)</li><li>Tran, Huy (PT 75)</li><li>r, r</li></ul>
<b>Lisätiedot</b> Jyrki Nummenmaa 040-5277999 jyrki.nummenmaa@gmail.com	<b>M-1850</b> <ul style="list-style-type: none"><li>Heiskanen, Mika (TIP-70)</li><li>Ojala, Matias (TuPy)</li><li>Risku, Jarkko (KoKu)</li><li>Faily, Jusef (PT 75)</li><li>Tanhua, Heikki (LPTS)</li><li>Holm, Veikko (PT 75)</li><li>Valtakoski, Jyri (PT 75)</li><li>Potiris, Rafail (PT 75)</li></ul>
<b>Toiminnot</b> <a href="#">Muokkaa tietoa</a> <a href="#">Lataa tiedot</a>	

Kuva 4: Turnauksen tiedot

## Muokkaa turnausta

Turnauksen tiedot	Turnauksen luokat																												
<b>Turnauksen nimi</b> PT 75:n kansalliset	<table><thead><tr><th>Nimi</th><th>Päivämäärä</th><th>Hinta</th><th>Poista</th></tr></thead><tbody><tr><td>MK</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr><tr><td>M-2000</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr><tr><td>M-1850</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr><tr><td>M-1700</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr><tr><td>M-1550</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr><tr><td>M-1400</td><td>22.02.2014</td><td>12,00</td><td><a href="#">Poista</a></td></tr></tbody></table>	Nimi	Päivämäärä	Hinta	Poista	MK	22.02.2014	12,00	<a href="#">Poista</a>	M-2000	22.02.2014	12,00	<a href="#">Poista</a>	M-1850	22.02.2014	12,00	<a href="#">Poista</a>	M-1700	22.02.2014	12,00	<a href="#">Poista</a>	M-1550	22.02.2014	12,00	<a href="#">Poista</a>	M-1400	22.02.2014	12,00	<a href="#">Poista</a>
Nimi	Päivämäärä	Hinta	Poista																										
MK	22.02.2014	12,00	<a href="#">Poista</a>																										
M-2000	22.02.2014	12,00	<a href="#">Poista</a>																										
M-1850	22.02.2014	12,00	<a href="#">Poista</a>																										
M-1700	22.02.2014	12,00	<a href="#">Poista</a>																										
M-1550	22.02.2014	12,00	<a href="#">Poista</a>																										
M-1400	22.02.2014	12,00	<a href="#">Poista</a>																										
<b>Turnauksen järjestävä organisaatio</b> PT 75																													
<b>Yhteyshenkilö</b> Jyrki																													
<b>Yhteyshenkilön puhelinnumero</b> 040-																													
<b>Yhteyshenkilön sähköpostiosoite</b>																													
<b>Alkamispäivä</b> 22.02.2014																													

Kuva 5: Turnauksen tietojen muokkaaminen

## Luo uusi turnaus

Turnauksen tiedot	Turnauksen luokat								
<b>Turnauksen nimi</b>	<table><thead><tr><th>Nimi</th><th>Päivämäärä</th><th>Hinta</th><th>Poista</th></tr></thead><tbody><tr><td colspan="4"><a href="#">Lisää luokka</a></td></tr></tbody></table>	Nimi	Päivämäärä	Hinta	Poista	<a href="#">Lisää luokka</a>			
Nimi	Päivämäärä	Hinta	Poista						
<a href="#">Lisää luokka</a>									
<b>Turnauksen järjestävä organisaatio</b>									
<b>Yhteyshenkilö</b>									
<b>Yhteyshenkilön puhelinnumero</b>									
<b>Yhteyshenkilön sähköpostiosoite</b>									
<b>Alkamispäivä</b>									

Kuva 6: Turnauksen luominen

Turnausluokat-välilehdeltä pystytään selaamaan, lisäämään ja muokkaamaan turnausluokkia. Turnaukset sisältävät aina tietyn määrän turnausluokkia, mihin pelaajat voivat ilmoittautua. Luokkiin voi olla myös erilaisia rajoitteita, kuten pelaajan rating-luku ja ikä. Turnausluokkien selaaminen, muokkaaminen ja lisääminen on esitetty kuvissa 7, 8 ja 9.

# PINGISMAISTERI

Turnaukset	Turnausluokat	Seurat	Oma tili	Kirjaudu ulos
------------	---------------	--------	----------	---------------

## Tallennetut turnausluokat

#	Luokan nimi	Rajoitteet	Lisätiedot	Toiminnot
5	MK		Poolit	<button>Muokkaa</button> <button>Poista</button>
6	M-2000	Rating: - 2000	Poolit	<button>Muokkaa</button> <button>Poista</button>
7	M-1850	Rating: - 1850	Poolit	<button>Muokkaa</button> <button>Poista</button>
8	M-1700	Rating: - 1700	Poolit	<button>Muokkaa</button> <button>Poista</button>

Kuva 7: Turnausluokkien selaaminen

## Muokkaa turnausluokkaa

Nimi

MK

Lisätiedot

Poolit

## Rajoitteet

Ikärajat



Kuva 8: Turnausluokan muokkaaminen

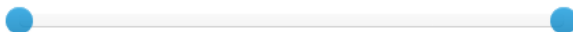
## Luo uusi turnausluokka

Nimi

Lisätiedot

## Rajoitteet

Ikärajat



Kuva 9: Turnausluokan luominen

Seurat-välilehdeeltä pystytään selaamaan ja hallinnoimaan seuroja. Nämä toiminnallisuudet on esitetty kuvissa 10, 11 ja 12.

## PINGISMAISTERI

Turnaukset	Turnausluokat	Seurat	Oma tili	Kirjaudu ulos
------------	---------------	--------	----------	---------------

### Seurat

ID	Nimi	Jäseniä	Poista
3	Atlas	2 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>
9	Ballong	0 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>
10	BF-78	0 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>
11	BK	0 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>

Kuva 10

14	Gurut	1 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>
20	Häki	3 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>
<hr/>			
	<b>Nimi</b>	<b>Toiminnot</b>	
	Pohjalahti,	<a href="#">Muokkaa tietoja</a>	
	Salo,	<a href="#">Muokkaa tietoja</a>	
	Virtanen,	<a href="#">Muokkaa tietoja</a>	
<hr/>			
15	HarSpo	0 <a href="#">Näytä jäsenet</a>	<a href="#">Poista seura</a>

Kuva 11: Seurojen selaaminen

## PINGISMAISTERI

Turnaukset	Turnausluokat	Seurat	Oma tili
------------	---------------	--------	----------

### Luo uusi seura

Name

  
[Tallenna](#)

Kuva 12: Seuran luominen

Oma tili –välilehdeltä pystytään hallinnoimaan omia käyttäjätietoja, sekä lisäämään ja hallinnoimaan omaan tiliin liitettyjä pelaajia. Nämä toiminnallisuudet on esitetty kuvissa 13 ja 14.

## PINGISMAISTERI

Turnaukset	Turnausluokat	Seurat	Oma tili	Kirjaudu ulos
------------	---------------	--------	----------	---------------

### Oma tili

#### Omat tiedot

Nimi Perttu  
Sähköpostiosoite perttu.hallikainen@uta.fi  
Puhelinnumero 040

#### Tiliin lisätyt pelaajat

Nimi	Osoite	Ikä	Sukup.	Seura	Yhteystiedot	Ilmoittautumiset
------	--------	-----	--------	-------	--------------	------------------

Kuva 13: Oma tili

## PINGISMAISTERI

Turnaukset	Turnausluokat	Seurat	Oma tili	Kirjaudu ulos
------------	---------------	--------	----------	---------------

Etinimi	<input type="text"/>
Sukunimi	<input type="text"/>
Syntymäpäivä	<input type="text"/>
Osoite	<input type="text"/>
Postinumero	<input type="text"/>

*Kuva 14: Pelaajan lisääminen omaan tiliin*

Turnauksen järjestäjän näkökulmasta järjestelmästä löytyy vielä lisää toiminnallisuuksia. Järjestäjä pystyy esimerkiksi arpomaan turnauspoolit sekä muokkaamaan turnauspooleja. Nämä toiminnallisuudet on esitetty kuvissa 15 ja 16.

### Poolien arvonta

Luokan nimi
MK
M-2000
M-1850
M-1700
M-1550
M-1400
Vet. avoin
M17GP
N17GP

*Kuva 15: Turnauspoolien arvonta*

Turnaukset	Turnausluokat	Seurat	Päivitä ratingit	Oma tili
------------	---------------	--------	------------------	----------

MK	M-2000	M-1850	M-1700	M-1550	M-1400
Vet. avoin	M17GP	N17GP	M14GP	N14GP	M12
TAS	Harrastelijat				

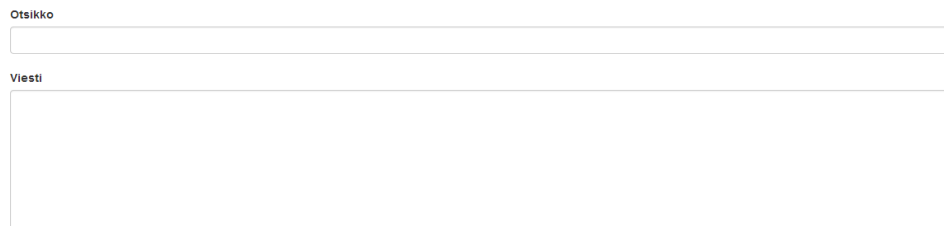
  

1	2	3
<b>Veikko</b> 1700 PT 75	<b>Heikki</b> 1677 LPTS	<b>Jarkko</b> 1618 KoKu
<b>Jyrki</b> 1591 Häki	<b>Mika</b> 1555 TIP-70	<b>Jyri</b> 1598 PT 75

*Kuva 16: Turnauspoolien muokkaaminen*

Turnauksen järjestäjän on myös mahdollista lähettää sähköpostia, turnauksiin osallistuville pelaajille. Sähköpostin lähettäminen on esitetty kuvassa 17. Pelaajien rating- tiedot pystytään päivittämään sptl.fi-sivustolta saatavasta Excel-tiedostosta. Päivittäminen on esitetty kuvassa 18.

Lähetä osallistujille sähköpostia



*Kuva 17: Sähköpostin lähettäminen osallistujille*

## PINGISMAISTERI



*Kuva 18: Rating –tietojen päivittäminen*

## Projektiryhmä ja projektin hallinnointi

### Projektipäälliköt:

Perttu Hallikainen

Jere Myyryläinen

Miikka Kähkönen

### Projektiryhmäläiset:

Laura Kautonen

Lauri Nykänen

Kimmo Ronkainen

Ryhmässämme oli alun perin kolme projektipäällikköä ja neljä projektiryhmäläistä. Yksi projektiryhmäläinen joutui kuitenkin jättämään projektin omalta osaltaan kesken kurssin alkuvaiheessa, joten ryhmän koko supistui kuuteen henkeen.

Projektipäälliköt olivat kaikki maisterivaiheessa opiskelevia 5-6. vuoden opiskelijoita ja kaikilla oli aikaisempaa kokemusta ohjelmistoalalta. Jere omasi vahvan Web-ohjelmointitaustan, Miikkalla oli työkokemusta peliohjelmoinnista ja Perttu on työskennellyt järjestelmänvalvojana ja ylläpitotiimin tiiminvetäjänä.

Muilla projektiryhmäläisillä ei ollut aikaisempaa työkokemusta ohjelmistoalalta, mutta aikaisempaa ohjelmointikokemusta löytyi. Laura on vuorovaikutteisen teknologian opiskelija, joten hänen vahvuusalueisiinsa projektissa kuului käyttöliittymäsuunnittelu, testaamisen suunnittelu ja testaamisen toteutus. Kimmo ja Lauri ovat tietojenkäsittelyopin opiskelijoita, joten ohjelmointi kuului pääasiassa heidän vastuulleen.

### **Projektin hallinnointi**

Projektin kehitysmallina toimi muokattu Scrum. Projekti jaettiin sprintteihin, joissa jokaiseen sprinttiin otettiin mukaan tietyt kehitystehtävät ja jokaisen sprintin tarkoituksena oli tuottaa toimiva komponentti. Kehitysmallia oli muokattu siksi, että tavallinen Scrum vaatisi tiiviimpää ryhmätyöskentelyä, kuin mihin ryhmällä oli resursseja. Käytännössä kehitysmalli toteutettiin siten, että joka viikko pidettiin viikkopalaveri. Viikoittaisten palaverien agendana olivat seuraavat asiat: mitä on tehty viime viikolla, mikä on projektin tila ja mitä tullaan tekemään seuraavaksi. Palavereiden välissä ryhmäläiset työskentelivät pääsääntöisesti itsenäisesti, mutta järjestimme kuitenkin usein yhteisiä ohjelmointitilaisuuksia. Kommunikaation apuvälineenä käytettiin irciä, Skypeä, sähköpostia ja Facebookia.

### **Menetelmät ja työkalut**

#### **Ohjelmointikieli**

Sovellus toteutettiin PHP -ohjelmointikielellä käyttäen CakePHP-ohjelmistokehystä. CakePHP mahdollisti projektissa hyvin nopean ohjelmistokehityksen, koska se sisältää vahvat ja valmiit toteutukset moneen yleiseen pulmaan.

#### **Koodin ja laadun analysointi**

Koodin ja laadun analysointiin valittiin aluksi Tampereen yliopiston tarjoamaa Jenkins-ohjelmisto. Ohjelmistolla oli tarkoitus tehdä järjestelmän yksikkötestaus, mutta jouduimme toteamaan kesken projektin, että ryhmän resurssit eivät riitä yksikkötestauksen tekemiseen.

#### **Käyttöliittymän ulkoasun toteutus**

Käyttöliittymän toteutukseen käytettiin Twitter Bootstrapia. Twitter Bootstrap tarjosi hyvän työkaluvalikoiman helppoon ja selkeään käyttöliittymän toteutukseen.

#### **Testaaminen**

Projektissa käytettiin niin sanottua asiantuntijatestausta, missä projektipäälliköt tekivät pääasiassa testaamisen. Projektin yhteydessä toteutettiin myös käytettävyytestisuunnitelma ja testit toteutettiin suunnitelman mukaisesti.

#### **Versionhallinta**

Projektin versiohallintaan käytimme Subversionia. SVN oli kytketty Redmine-projektinhallintajärjestelmään.

#### **Tietokanta**

Sovelluksen tietokantaohjelmistona toimii MySQL. MySQL-tietokantaa hallinnoitiin phpMyAdmin-sovelluksella.

#### **Projektin hallinta**

Projektin hallintatyökaluna käytettiin Tampereen Yliopiston tarjoamaa Redmine-ohjelmistoa. Redmineen kirjattiin muun muassa ryhmäläisten tunnit ja työtehtävät, joiden perusteella raportointi tapahtui. Dokumenttien ja vaatimusten hallinta tapahtui Redminen kautta.

## **Yhteisesti toteutettavat dokumentit**

Yhdessä toteutettavat dokumentit tehtiin Google Driven työkaluja käyttämällä.

## **Käyttöliittymäsuunnittelu**

Käyttöliittymäsuunnittelu tehtiin Tampereen Yliopiston tarjoamaa Balsamiq –ohjelmistoa hyödyntämällä.

## **Kehitysympäristö**

Kehitysympäristöstä ei ollut yhdenmukaista linjausta vaan jokainen sai käyttää haluamiaan työkaluja. Projektipäälliköiden suosittelema kehitysympäristö oli Eclipse.

## **Kommunikaatio**

Kommunikointivälineiksi valittiin alun perin Skype, Irc ja sähköposti. Sähköpostilla hoidettiin viralliset tiedotteet ja tärkeimmät viestinnät. Muut työkalut tarjosivat reaaliaikaisemman kommunikointimahdollisuuden. Myöhemmin projektissa otimme käyttöön myös Facebookin.

## **Projektin vaiheet ja kehitysmalli**

Kehitysmallina toimi muunneltu Scrum. Malli poikkesi normaalisti Scrumista siten, että pidimme viikoittain tapaamisen missä kävimme läpi yhteiset asiat, toteutetut tehtävät ja seuraavat tehtävät. Muuten pidimme tiiviisti yhteyttä sähköisesti IRC:n, Facebookin ja Skypen kautta. Järjestimme myös jonkin verran yhteisiä ohjelmointitilaisuuksia, mutta nämä eivät olleet säännöllisiä. Kimmo ja Lauri tekivät jonkin verran ohjelmointityötä oma-aloitteisesti yhdessä.

## **Katselmoinnit:**

- 26.9: Projektiryhmä, molemmat asiakkaan edustajat ja kurssin vastuuhenkilö
- 18.10: Projektiryhmä, asiakkaan edustaja ja kurssin vastuuhenkilö
- 27.11 Projektiryhmä, molemmat asiakkaan edustajat ja kurssin vastuuhenkilö
- 30.1 Projektiryhmä, toinen asiakkaan edustaja
- 19.2 Projektiryhmä, toinen asiakkaan edustaja ja kurssin vastuuhenkilö
- 12.3 Projektiryhmä, asiakkaan toinen edustaja ja kurssin vastuuhenkilö

## **Dokumentit**

- 25.9 Preliminäärialalyysi
- 17.10 Projektisuunnitelma
- 12.3 Projektitarina
- 12.3 Loppuraportti



### Toteutunut vaiheistus

<b>Sprint 0</b>		<b>175,75 / 150</b>
Tavoite: 04.09. - 07.10.	Projektisuunnitelma	OK
<b>04.09. - 07.10.</b>	Esianalyysi	OK
	Työkalujen opiskelu	Jatkui kurssin ajan
<b>Sprint 1</b>		<b>72,25 / 80</b>
Tavoite: 8.10. - 21.10.	Uuden turnauksen luominen	OK
<b>08.10. - 28.10.</b>	Turnaukseen ilmoittautuminen	OK
	<b>Pelaajien luonti (uusi)</b>	OK
	<b>Käyttöliittymät turnauksen hallinnointi ja ilmoittautumis osioon (uusi)</b>	OK
<b>Sprint 2</b>		<b>154,5 / 150</b>
Tavoite: 21.10. - 11.11.	Poolien arvonta	Ei aloitettu
<b>28.10. - 10.12.</b>	Ratingin ja pelaajien parsetus	Ei valmistunut
	Tuloksien kirjaaminen	Ei aloitettu
<b>Sprint 3</b>		<b>67 / 150</b>
Tavoite: 12.11. - 2.12.	Tuloksien tulostaminen	Ei aloitettu
10.12. - 07.01.	Dokumenttien tulostaminen	Ei aloitettu
	Liiton vaatima data	Ei aloitettu
	Turnaustiedon lataus	Ei aloitettu
	Personal report 1	OK
	<b>Poolien arvonta (uusi)</b>	Ei valmistunut

	<b>Ratingin ja pelaajien parsetus (jatkuu)</b>	Ei valmistunut
	<b>Käytettävyytestisuunnitelma (uusi)</b>	Ei valmistunut
<b>Sprint 4</b>		<b>179,25 / 150</b>
Tavoite: 3.12. - 6.01.	Turnauksien hallinta	OK
07.01. - 04.2.	Luokkien luonti ja hallinta	Tehty sprint 1
	Pelaajien selaus	Tehty sprint 1
	Ilmoittautumisten hallinta	Tehty sprint 1-3
	<b>Poolien arvonta (jatkuu)</b>	Ei valmistunut
	<b>Turnaustiedon paketointi (uusi)</b>	OK
	<b>Käytettävyytestisuunnitelma (jatkuu)</b>	OK
	<b>Ratingin ja pelaajien parsetus (jatkuu)</b>	Ei valmistunut
<b>Sprint 5</b>		<b>138 / 100</b>
Tavoite: 07.01 - 27.01.	Pöytien hallinta	Ei toteutettu
04.02. - 21.02	Käyttäjärekisteri	Sprint 1
	Käyttäjien hallinta	Sprint 1
	<b>Käytettävyytestit (Vain online osio)</b>	OK
	<b>Poolien arvonta</b>	OK
	<b>Ratingin ja pelaajien parsetus (jatkuu)</b>	OK
	Personal report 2	OK
<b>Sprint 6 (jätettiin pois)</b>		<b>- / 150</b>

28.01. - 17.02.	Käytettävyyssanalyysi	Ei tehdä
	Käytettävyyden parantaminen	-
	Käyttäjättestit	Ei tehdä
<b>Sprint End</b>		<b>53 / 50</b>
Tavoite: 17.02. - kurssi loppuun	Loppuraportti	OK
21.02 – kurssi loppuun	Personal report 3	OK
	Käytettävyyden parantaminen	OK
	<b>Pilotista esiintyneiden huomioiden korjaaminen (uusi)</b>	OK
	<b>Tuloksien kirjaaminen (uusi)</b>	OK

## Kokemuksia

Projektissa ei kohdattu suuria vaikeuksia tai ongelmia. Osa ennakoituista riskeistä kuitenkin toteutui, mutta ne eivät estäneet projektin toteutusta. Merkittävimmät toteutuneet riskit liittyivät projektin työtuntiresursseihin kun yksi ryhmäläinen joutui jättämään projektin kesken kurssin alkuvaiheessa ja kahden ryhmäläisen työtuntiarviota jouduttiin tiputtamaan 40 tunnilla. Tästä toki seurasi osittain myös muiden riskien toteutumista, mutta riskeistä toivuttiin tehtävien ja resurssien uudelleen priorisoimisella. Projektissa ei kohdattu ennakoimattomia riskejä. Riskien toteutumisen vaikutus lopputulokseen oli lähinnä se, että kaikkia niitä osioita ei voitu ottaa mukaan projektiin, mitä aluksi suunniteltiin. Tästä myös aiheutui pienehkö projektin venyminen. Riskit on eritelty tarkemmin seuraavassa taulukossa:

### Riskiasteikon kuvaus

Todennäköisyys ilmaisee sitä, kuinka todennäköiseksi kyseinen riski oletettiin projektin alussa. Riskit on luokiteltu vakavuusjärjestyksen mukaan asteikolla 1-5 siten, että 1 on epätodennäköinen ja 5 on todennäköinen

### Riskien vakavuus

Riskien vakavuusaste on kuvattu asteikolla 1-5: 1 Tarkoittaa erittäin lievää riskiä ja 5 tarkoittaa erittäin vakavaa riskiä. Alla tarkempi kuvaus asteikosta:

**(5) Erittäin vakava.** Riskin toteutuminen vahingoittaa koko projektia ja voi estää koko projektin etenemisen.

**(4) Vakava.** Riskin toteutuminen haittaa projektia ja hidastaa sen etenemistä sekä voi

estää projektin osa-alueiden toteutumisen.

**(3) Keski-verta.** Riskin toteutuminen hidastaa projektin valmistumista ja joistain toiminnallisuuksista voidaan joutua tinkimään.

**(2) Lievä.** Riskin toteutuminen aiheuttaa projektiin lisätyötä, mutta projekti pysyy aikataulussa ja halutut toiminnot saadaan toteutetuksi.

**(1) Erittäin lievä.** Riskin toteutuminen aiheuttaa ryhmälle pientä lisätyötä.

ID	Kuvaus	Toden näköisyys	Va-ka-vuus	Ennalta ehkäisy	Torjunta	Toipuminen	Oire	Toteutuminen
1	Tuntematon tekniikka. Uudet työskentelytavat, ohjelmointikieli ja muu tekniikka	4	3	Laaja perehtymisvaihe projektin alussa, jonka aikana ryhmä opettelee uusien tekniikoiden käytön	Projektipäälliköt auttavat tarvittaessa. Projektipäälliköt pystyvät tarvittaessa tukemaan ongelmia aiheuttavien kohtien toteutuksessa	Kun ongelma on ratkaistu, lisätään panostusta projektiin. Ongelman ratkaisu kerrataan koko ryhmälle, jotta vastaava tilanne voidaan ohittaa nopeammin jatkossa.	Työn laatu on heikkoa ja projekti ei etene halutulla tavalla.	Toteutui osittain. Ohjelmointikieli oli suurimmalle osalle entuudestaan tuntematon, joten projekti vaati paljon opiskelua ja tietyt tehtävät jäivät välillä jumiin, koska tarvittiin projektipäälliköiden asiantuntemusta, mitä ei välttämättä ollut heti saatavilla. Torjunta kuitenkin auttoi ja alussa panostettiin opiskeluun.
2	Projektille ei riitä tarpeeksi aikaa muiden henkilökoh- taisten velvollisuuksien takia	4	3	Projektin aikataulussa otetaan huomioon ryhmäläis-ten mahdollisuuksien panostaa projektiin.	Jokainen ryhmäläinen varaa viikosta vaaditun ajan projektille. Toteutettavat kokonaisuudet pyritään määrittämään siten, että vastuuhenkilön aika riittää vastualueen suorittamiseen	Panostuksen lisääminen kun aikaa löytyy taas muilta töiltä. Varsinkin joululoman aikana on mahdollista ottaa kiinni rästiin jääneitä tehtäviä. Aikaa tulee kuitenkin jäädä myös lepoon.	Tehtävät jäävät tekemättä. Väsyminen projektiin. Tehtävien laatu heikkenee.	Riski toteutui, koska kahden projektiryhmäläisen työmääräarviota jouduttiin pudottamaan. Riskin toteutumisesta toivuttiin ajankäytön ja työtehtävien uudelleenpriorisoinnilla, joten suurta haittaa projektille ei aiheutunut.
3	Projektin aikataulutus on tehty liian tiukaksi.	2	4	Projektin vaiheiden toteuttamiseen varataan tarpeeksi aikaa.	Ryhmäläiset noudattavat saamiaan aikarajoja ja ilmoittavat ajoissa, jos tilanne vaikuttaa huonolta.	Aikataulua tarkistetaan ja määritellään mitkä ominaisuudet ovat tärkeimpiä ja vähemmän tärkeiden ominaisuuksien toteutus siirretään projektin loppuun	Aikataulu on kokoajan myöhässä. Tehtävät eivät valmistu ajallaan.	Toteutui osittain riski ID 2:n toteutumisesta johtuen. Ei kuitenkaan aiheuttanut suurta haittaa projektille.
4	Motivaation puute	2	4	Projektipäälliköt pyrkivät omalla esimerkillään pitämään ryhmän kiinnostuksen yllä.	Huonosti motivoituneen ryhmän jäsenen kanssa keskustellaan ja pyritään ohjaamaan tätä kiinnostaviin tehtäviin.	Yritetään siirtää työtehtäviä siten, että kaikki saisivat tehdä itselleen mielekkäitä tehtäviä.	Tehtävien tekemättä jättäminen ja siitä ilmoittaminen kärsii.	Ei toteutunut.
5	Ryhmän jäsenen sairastumi-	2	3	Tehtäviä voi tehdä myös ennakoon	Jos kyseessä on pitempiaikainen sairaus/poissaolo	Sairastelun/poissaolon jälkeen panoksen	Ryhmän jäsenen katoa, eikä ilmoita	Toteutui. Yksi ryhmän jäsenistä joutui keskeyttämään

	nen tai muu pitkä poissaolo			jos tiedossa on esim. matka.	ilmoitetaan siitä muulle ryhmälle, jotta tehtävät voidaan uudelleen organisoida.	lisääminen projektiin ja omien tehtävien tunnollinen suorittaminen	asiasta. Projektin aikataulu myöhästyy ja tehtäviä voi jäädä kokonaan tekemättä.	projektin omalta osaltaan henkilökohtaisista syistä. Riskin toteutumisesta toivuttiin tehtävien ja ajankäytön uudelleenpriorisoinnilla.
6	Lopputuote on liian vaikea käyttää	1	3	Käyttöliittymäsunnittelu on käytetään tarpeeksi aikaa. Käytetään valmiita käyttöliittymäkomponentteja.	Käytettävyyteen keskitytään kun käyttöliittymäsunnittelu tulee ajankohtaiseksi. Myös arkkitehtuurin suunnittelulla voi olla vaikutusta käytettävyyteen.	Kysytään loppukäyttäjiltä miten ongelma voitaisiin ratkaista. Tutkitaan muita järjestelmiä, joissa samankaltaisia ongelmia on ratkaistu.	Asiakas ei selviä järjestelmän käytöstä yksinään vaan tarvitsee projektiryhmän tukea tai kattavan ohjeistuksen.	Ei toteutunut.
7	Asiakkaan asettamat vaatimukset projektille kasvavat liian suuriksi	1	2	Asiakas pidetään mukana kommunikointiossa. Samalla korostetaan sitä, ettei projektin vaatimusmäärittely ole ns. elävä dokumentti vaan siihen tulisi tulla mahdollisimman vähän muutoksia projektin aikana.	Uusien vaatimuksien ilmetessä pyritään keskustelemaan asiakkaan kanssa ja tuodaan esille ryhmän näkökulma projektin vaatimuksien muutokselle.	Pyritään pilkkomaan toteutettavaksi valitut toiminnallisuudet tarpeeksi pieniin osiin, jolloin niiden toteutus ei ole yhden henkilön harteilla.	Tasaiseen tahtiin tulevat muutos- tai lisäystoiveet projektin suhteen.	Ei toteutunut. Asiakkaan kanssa pysyttiin selkeässä linjassa siitä, mitä pitää ja mitä pystytään toteuttamaan resurssien puitteissa.

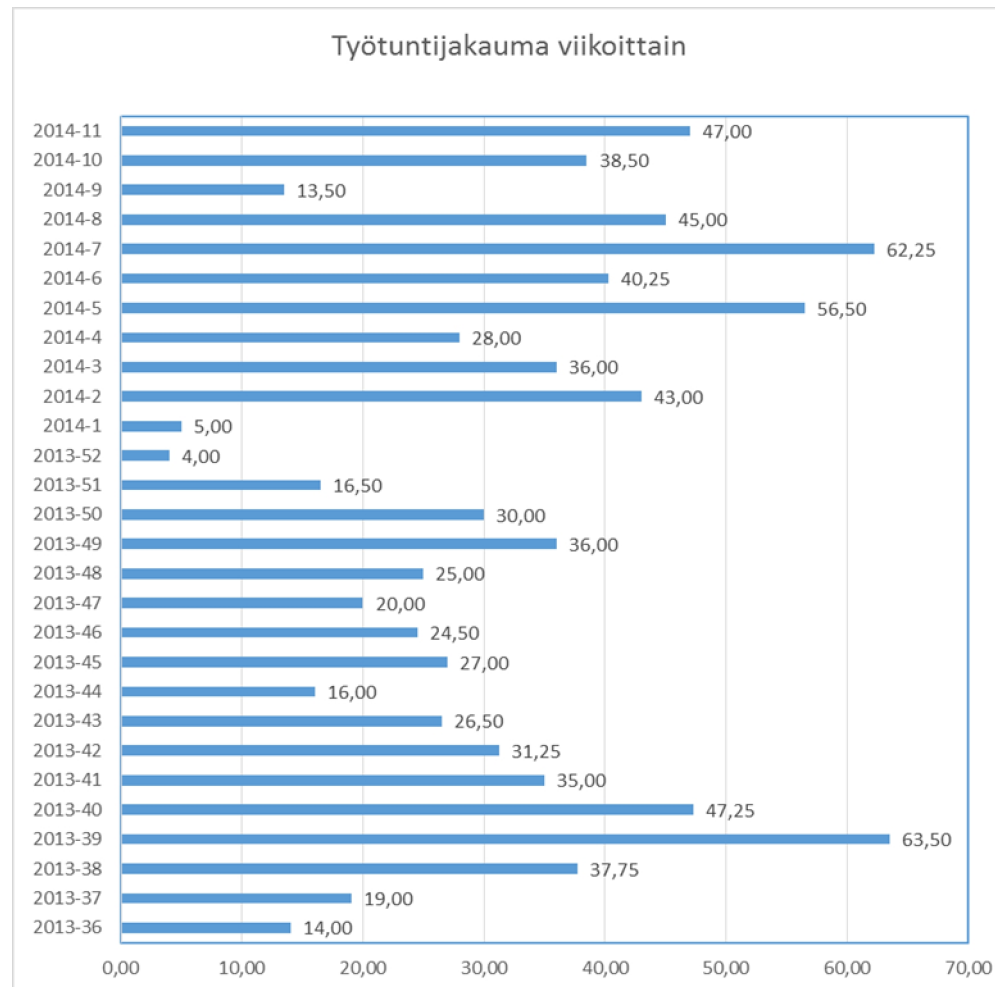
### Kokemukset projektista

Ryhmämme mielestä projekti onnistui hyvin. Saimme aikaan toimivan järjestelmän, josta on hyötyä asiakkaalle. Valtaosa sovitusta toiminnallisuudesta onnistuttiin myös toteuttamaan määritellyllä tavalla.

Projekti oli mielestämme myös siinä mielessä onnistunut, että ryhmässämme oli erittäin hyvä tiimihenki ja kaikki tekivät tasavertaisesti töitä omiin resursseihinsa nähden. Ryhmän kanssa oli mukava työskennellä ja kaikilla oli hyvä asenne projektiin.

Projekti oli myös oppimisen kannalta hyödyllinen. Projektipäälliköt saivat arvokasta kokemusta projektin hallinnoinnista ja johtamisesta. Monille ryhmäläisille tämä oli ensimmäinen kokemus varsinaisesta ohjelmistoprojektista ja he saivat hyvää kokemusta ohjelmoinnista ja projektiluontoisesta työskentelystä. Projekti auttoi varmasti myös hahmottamaan kokonais kuvaa siitä, mitä tämän tyyppiseen projektiin kokonaisuudessaan kuuluu.

## Tilastot



Tiimin koko	Kehitysmalli	Aloituspvm.	Lopetuspvm.	Päiviä	Tunteja
3+3	Muunneltu Scrum	9.9.2013	12.3.2014	184	889.25

*Taulukko 1: Yleiset projektitiedot.*

Aktiviteetti	Suunnittelu ja hallinnointi	Vaati musten määrittely	Design	Koodi	Integrointi ja testaaminen	Katselmoinnit	Korjauks	Opiskelu	Muu	Yhteensä
Tunnit	333,75	21,5	29,5	269,75	48	24,50	10,75	99,5	69	906,25
%	37%	2%	3%	30%	5%	3%	1%	11%	8%	100%

*Taulukko 2: Jakauma aktiviteettien mukaan*

Vaatimukset	Sivut	Käyttötapa- ukset	Käyttöliittym- äsivut	Tietokantaka- aviot	Tietokantatau- lut
21	3	15	21	3	15

*Taulukko 3: Vaatimukset ja korkean tason suunnittelu*

Sivut	Yleiskuvaka- viot	Luokkakaavi- ot	Sekvenssikaa- viot	Tilakaaviot	Muut kaaviot
14	3	2	0	4	5

*Taulukko 4: Suunnittelu*

Dokumentti	Sivut	Versiot
Preliminäärianalyysi	7	2
Projektisuunnitelma	22	15
Käytettävyytestiraportti	9	3
Vaatimukset	3	9
Testisuunnitelma	7	3
Loppuraportti	26	11
Projektitarina	14	2
Viikkoraportit	24	24

*Taulukko 5: Dokumentit*

Language	PHP
LOC	2592
SLOC	2160
Reused code	104
Reused and modified	0
Classes	25
Functions	69
Code revisions	154

*Taulukko 6: Koodi*

# Games for Elementary Mathematics Learning Environment

## Overview

Games for elementary mathematics learning environment is a software project designed for primary school students whose ages vary between 6 and 9. The product contains different games from different mathematical topics at the level of targeted student profile. Therefore, the product aims to teach and practice mathematical subjects in more entertaining and interactive way. The latest research has proven that games can be considered as learning tools since they provide activity and interaction. Then, it can be interpreted that the product will be beneficial for learning of the students.

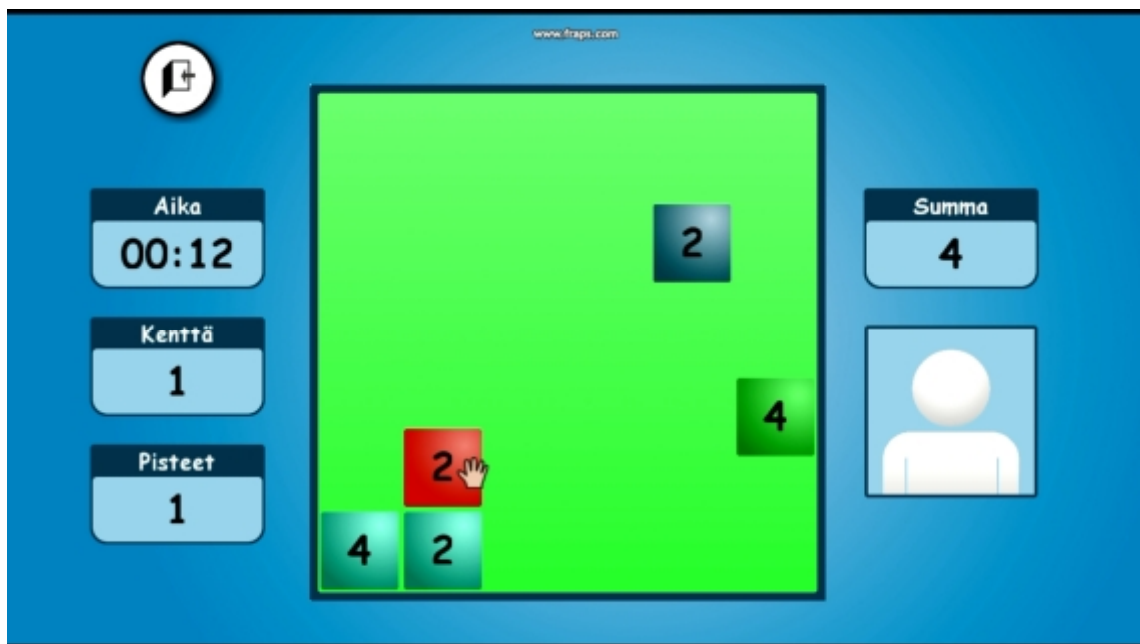
The product uses Kinect technology. So that, games are being controlled by movements of hands. At the beginning of the project, there were three games implemented about quantities of objects, numerical symbols and basic mathematical arithmetic. All previously implemented games were based on holding an object and leaving it to the correct place. Basically, four games were implemented in this version. These games can be divided into two categories as cube and balloon games. The main idea of cube games is to reach a sum shown on the screen by holding a numbered cube and dragging it onto another numbered cube. Display of the sum varies in these two games. Therefore, sum can be shown with numbers or dots. The main idea of the balloon games is to find target numbered balloon and put it into the bin. Children are supposed to reach result of a calculation or find missing number in a pattern.

## GUI Screenshots

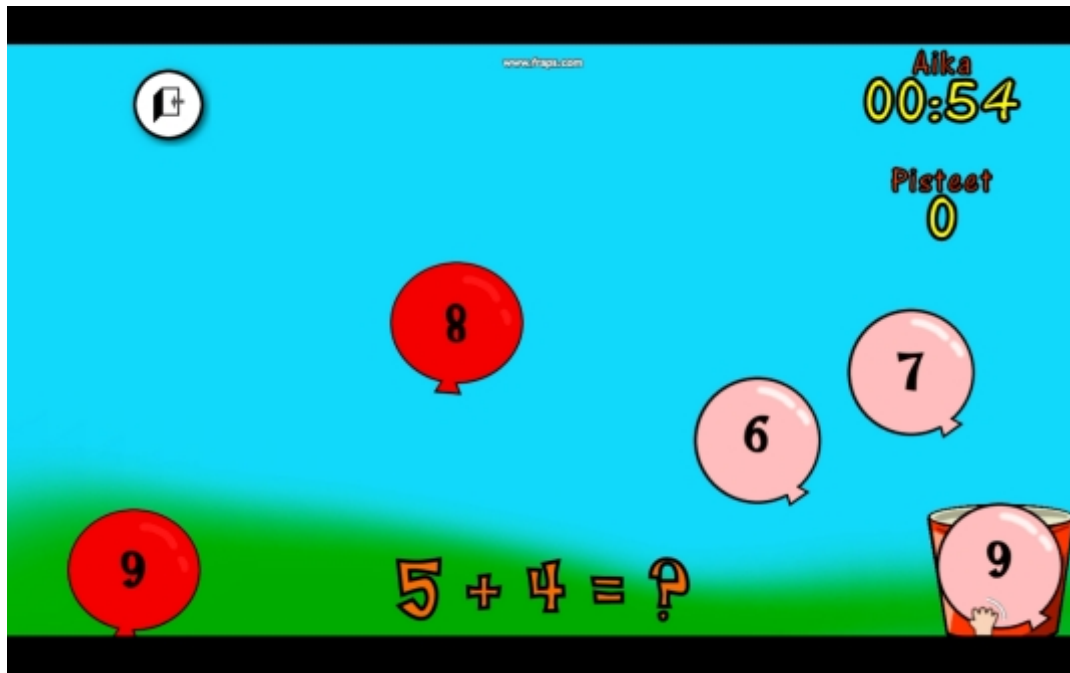


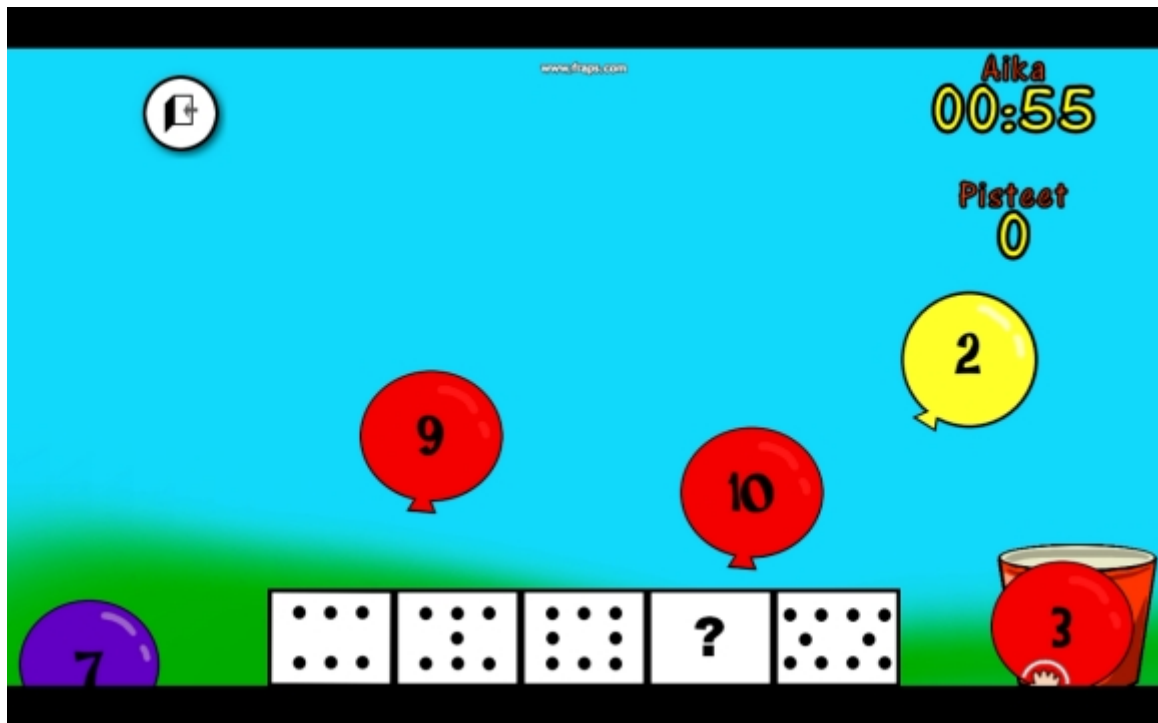












## Organization and management

### Group



Figure 1: Group photo

#### Batuhan Baykara

- **Knowledge:** Java, C, C++, PHP, SQL, Assembly Language, Lex&Yacc, HTML, MATLAB, JSF, SPRING Framework, PrimeFaces, Scheme, Octave, Verilog, Microsoft Visual Studio, Eclipse, Netbeans, JCreator, XAMPP Server, Apache Tomcat, RazorSQL

#### Dilem Aydınli

- **Knowledge:** Java, C, C++, SQL, MATLAB, Lex & Yacc, PHP, Scheme, MIPS, Microsoft Visual Studio, Eclipse, NetBeans, JCreator, Adobe Flex, Photoshop, Flash

#### Ruibin Ye

- **Knowledge:** Java, C, C++, SQL, PHP, SOAP/REST Web service, Visual Studio, Java Server Faces, Java Persistence API, Servlet, iOS development, SQLite, MySQL, Google app engine, Oracle, DB2, Visual Studio, Eclipse, Netbeans, Tomcat, Xcode.

#### Philipp Weitz

- **Knowledge:** C, C++, Java, Javascript, JSP, Shell scripts (ksh, bash), Liferay, Vaadin, Spring, Java persistence API, MySQL, PostgreSQL, CSS

**Mykola Andrushchenko**

- **Knowledge:** Delphi/Pascal, C++, C#, Web Design (HTML/CSS, JavaScript and PHP) , database programming with MS SQL Server

**Tsailing Wong**

- **Knowledge:** C++, C#, JAVA

**Zhaodong Fan**

- **Knowledge:** C, C++, Low level programming

**Client**

Information about clients can be seen in Table 2. Regular meetings with clients were held through the project.

Name & Surname
Professor Markku Turunen
Jussi Okkonen

**Table 2: Information of clients**

**Supervisor**

The information of project supervisor can be found in Table 3.

Name & Surname
Pekka Mäkiaho

**Table 3: Information of the supervisor**

**Methods and Tools****Tools**

In the project C# programming language was used and Microsoft Visual Studio 2010 was the main programming tool in the project. In order to run the application, the environment need the distributable packages that comes with Microsoft Visual Studio 2010 and all the C++ distributable packages. Furthermore, XNA Game Studio 4.0, Kinect SDK 1.7 and skeleton server were other environment constraints before running the application.

The project built around OOP structure which makes it very easy to communicate through objects in the system. Also the system uses MVC as its software architecture and design pattern. This architecture makes the implementation easier since every new added feature, exercise and game in the system fall into its place easily.

The project was using SVN server in SIS. We used subversion to maintain current and



historical versions of source code and documentation. Also the subversion is connected to Redmine and Jenkins. However, we weren't be able to use Jenkins.

Our group used project wiki and it running. As project management tools, we used Google Drive to keep and complete our documentations. All the testing was planned to be done in Visual Studio. Lastly we had unit tests right after we finish implementing major components and afterwards we did system tests on the product.

## **Project phases and development model**

### **Development Model**

As the project's development model, an agile development model has been chosen which is customized scrum. In the project there were 7 sprints and each sprint consisted of 3 weeks. A backlog was held throughout the project and it updated all the time when there is a change in requirements. Work phases can be found in Appendix A.

### **Milestones**

<b>Deliverables</b>	<b>Due Dates</b>	<b>Done</b>	<b>Group in Charge</b>
Project is assigned. Meeting with clients and team members	04.09.2013	Done	Management and Developer Groups
Weekly Report	Every Week	Done	Management Group
Preliminary Analysis Report	27.09.2013	Done	Management Group
Project Plan Document	05.10.2013	Done	Management Group
UI and Sound Design	15.10.2013	Done	Developer Group
User Stories	Continuous	Done	Management Group
Project inspection and meeting minutes	10.10.2013	Done	Management and Developer Groups
Review Meeting 1 (Clients and Supervisor) and meeting minutes	29.11.2013	Done	Management and Developer Groups
Personal Reports 1	07.11.2013	Done	Management and Developer groups



Midterm Presentation	04.12.2013	Done	Management Group
Review Meeting 2 (Clients and Supervisor) and meeting minutes	12.12.2013	Done	Management and Developer groups
Personal Reports 2	14.01.2014	Done	Management and Developer groups
Review Meeting 3 (Clients and Supervisor) and meeting minutes	27.01.2014	Done	Management and Developer groups
Final report	04.03.2014	Done	Management Group
Final test report	04.03.2014	Done	Management Group
Final review meeting	04.03.2014	Done	Management and Developer groups
Final Presentation	05.03.2014		Management and Developer groups
Project CD	07.03.2014		Management Group

## Experiences

### Experience and What to Do Better Next Time

The project is successful as client is satisfied with our result and the project process according schedule. The team also gain lots of experience in the project.

Developers learns how to be a developer in a project team. They also get themselves familiar with C#, XNA, Kinect development environment. In the multicultural team, the conscious of cooperability was cultivated.

Managers learns how to be a manager in a project team. Since our team is using scrum development model, they put the theory into practice and experience how this model facilitate the development process. They also gained experience in software project management.

Project team should be aware of testing issues as early as possible or they might get hurry in later phases of the project.

In requirement engineering, client involvement is very important. Without client involvement, the project will lose direction at some points. In Scrum, user stories should always be updated. Use of Scrum development tool may help a lot.

### Foreseen Risks

Foreseen risks which were described in Risk Management section in Project Plan report did not show up during the schedule. For schedule factors, low risk evidence were encountered for all possible risks. The project started and completed on time, developer team followed the schedule and the project was parallel with the schedule. Deliverables were sent on time. required resources were acquired by project managers on time.

For requirements factors, requirements were clear, understandable and applicable and didn't cause falling behind of the schedule. Requirements were completed however, some of requirements needed to be rejected later on.

For personnel factors, team members were always available in required times. Although, group members didn't have any similar application experience before, they avoided schedule changes by self studying and helping each other. Therefore, team members worked cooperatively.

For the case of customer/user factors, clients involved in needed phases, performed a end product test and attended all review meetings.

Although this was the first project management experience of project managers, they were able to make the project completed on time.

### Unforeseen Risks

Through the end of the project, all requirements of MathNinja game were rejected by project managers after approval of the client. The reason behind that, developers were already tired to deal with putting so much effort on previous games. Therefore, MathNinja game were changed with two mini games. Overall, there are 4 games instead of 3 as it was planned before.

### Statistics

Team size	Dev. model	Start date	End date	Days	Hours
3+4	Scrum	27.09.2013	04.03.2014	158	953

*Table 1: General project information.*

Acti- vity	Planning and managem ent	Req. specifica tion.	De- sign	Code	Integratio n and testing	Revie ws	Repair	Study	Other	Total
Hours	511	15	49.5	159	86.5	14	12	80.50	25.5	<b>953</b>
%	53.6	1.57	5.19	16.68	9.07	1.46	1.26	8.44	2.67	100%

Table 2: Group effort by activity.

Number of user stories	Pages	UI screens	Database diagrams
89	4	6	2

Table 3: Requirements and high-level design outcomes.

People	Estimated hours	Actual hours	Difference
Ruibin Ye	175	141.00	34
Philipp Weitz	200	141.50	58.5
Dilem Aydinli	175	141.00	34
Batuhan Baykara	175	150.50	24.5
Zhaodong Fan	200	120.00	80
Mykola Andrushchenko	150	109.00	41
Tsailing Wong	200	150.00	50
Total	1275	953.00	322

Table 4: Resource plan and reality.

Document	Pages
Preliminary analysis	12
Project Plan	36
User Stories	4
Test plan	15
Test report	15
Final report	17
Project's story	19
Weekly reports	23
Review Meeting Reports	3

Table 5: Documents.

Language	C#
LOC	11466
SLOC	16467
Classes	152
Functions	865
Comments	1693
Code revisions	89

Table 6: Codelines (previous project included).



Table 7 : Repository statistics

## Appendix A - Work Phases

Phases	Tasks	Due date	Corresponding person
<b>Sprint 1</b>	Project Initiation <ul style="list-style-type: none"> <li>o Identify tool requirements</li> <li>o Acquire required tools</li> </ul>	27.9.2013	Whole team
	Meetings <ul style="list-style-type: none"> <li>o Project team meetings</li> <li>o Client meetings</li> <li>o Supervisor meetings</li> </ul>		Whole team
	Preliminary Analysis <ul style="list-style-type: none"> <li>o Analyze current product <ul style="list-style-type: none"> <li>§ Analysis of previous reports and code</li> </ul> </li> <li>o Analyze initial risks</li> <li>o Analyze initial requirements <ul style="list-style-type: none"> <li>§ Analyze functions</li> <li>§ Elicit general requirements</li> <li>§ Come to an agreement on the requirements with the clients</li> </ul> </li> </ul>	27.9.2013	Managers
	Documentation <ul style="list-style-type: none"> <li>• Weekly reports</li> </ul>	27.9.2013	Managers
	Project plan and project plan inspection	11.10.2013	Whole team
	Adapting to C# and Kinect environment	6.10.2013	Developers

	Design of software <ul style="list-style-type: none"> <li>o User interface improvement of main menu</li> <li>o Sound improvement of main menu</li> </ul>	6.10.2013	Developers
<b>Sprint 2</b> October 7, 2013 October 27, 2013	Installation and initialization tests	11.10.2013	Developers
	Requirements and Design of software <ul style="list-style-type: none"> <li>o Elicit requirements and perform architectural design of the games</li> <li>o Validate and rearrange the software architecture of previous design</li> </ul>	18.10.2013	Developers
	Other Designs <ul style="list-style-type: none"> <li>o User interfaces <ul style="list-style-type: none"> <li>§ UI Plan</li> </ul> </li> <li>o Sound</li> </ul>	18.10.2013	Developers
	Implementation <ul style="list-style-type: none"> <li>● Rearrange previous architecture</li> <li>o Code user interfaces</li> <li>o Sound</li> </ul>	27.10.2013	Developers
	Meetings <ul style="list-style-type: none"> <li>o Project team meetings</li> <li>o Review meetings <ul style="list-style-type: none"> <li>● Validating implementation and requirements with clients</li> </ul> </li> </ul>	27.10.2013	Whole team
	Documentation <ul style="list-style-type: none"> <li>● Weekly reports</li> <li>● Test reports</li> <li>● Requirements and Design Documentation</li> </ul>	27.10.2013	Managers
<b>Sprint 3&amp;4</b> October 28, 2013 December 8, 2013	Testing <ul style="list-style-type: none"> <li>o User interface <ul style="list-style-type: none"> <li>§ UI tests</li> </ul> </li> <li>o Sound tests</li> </ul>	30.10.2013	Developers
	Meetings <ul style="list-style-type: none"> <li>o Project team meetings</li> <li>o Review meetings</li> <li>o Validating implementation and</li> </ul>		Whole team

	requirements with clients		
	<ul style="list-style-type: none"> <li>● Implementation <ul style="list-style-type: none"> <li>○ Balloon game with numbers</li> <li>○ Cube game with numbers</li> </ul> </li> </ul>	08.12.2013	Developers
	Midterm presentation and preparation	04.12.2013	Managers
	Documentation <ul style="list-style-type: none"> <li>● Weekly reports</li> <li>● Personal report I</li> <li>● Test reports</li> </ul>	07.11.2013	Managers
<b>Sprint 5&amp;6</b> December 9, 2013	Testing Previous games	12.12.2013	Developers
	Requirements and Design <ul style="list-style-type: none"> <li>● Elicit requirements</li> <li>● Design UI of games with patterns</li> <li>● Design UI of Sound</li> </ul>	19.12.2013	Whole Team
	Implementation <ul style="list-style-type: none"> <li>● Balloon and cube game with pattern</li> <li>● Continue the other games if necessary</li> </ul>	19.01.2014	Developers
	Meetings <ul style="list-style-type: none"> <li>○ Project team meetings</li> <li>○ Review meetings</li> <li>○ Validating implementation and requirements with clients</li> </ul>		Whole team
	Documentation <ul style="list-style-type: none"> <li>● Weekly reports</li> <li>● Personal report II</li> <li>● Test reports</li> </ul>	15.01.2014	Managers
<b>Sprint 7</b> February 3, 2014	Testing All the Games	10.02.2014	Whole Team
	Meetings <ul style="list-style-type: none"> <li>○ Project team meetings</li> <li>○ Review meetings</li> <li>○ Validating implementation and requirements with clients</li> </ul>		Whole team

	Documentation <ul style="list-style-type: none"> <li>• Weekly reports</li> <li>• Test reports</li> </ul>		Managers
	Perform integration		Developers
<b>Sprint 8</b> 7, March 2014	2 weeks		
	Finalizing the project	07.03.2013	Whole Team
	Team meeting		Whole Team
	Final meeting	07.03.2013	Whole Team
	Prepare final deliveries	07.03.2013	Whole Team
	Documentation <ul style="list-style-type: none"> <li>• Weekly reports</li> <li>• personal report III</li> </ul>		

# Time Distance Speed Experiment for High School Students

## Overview

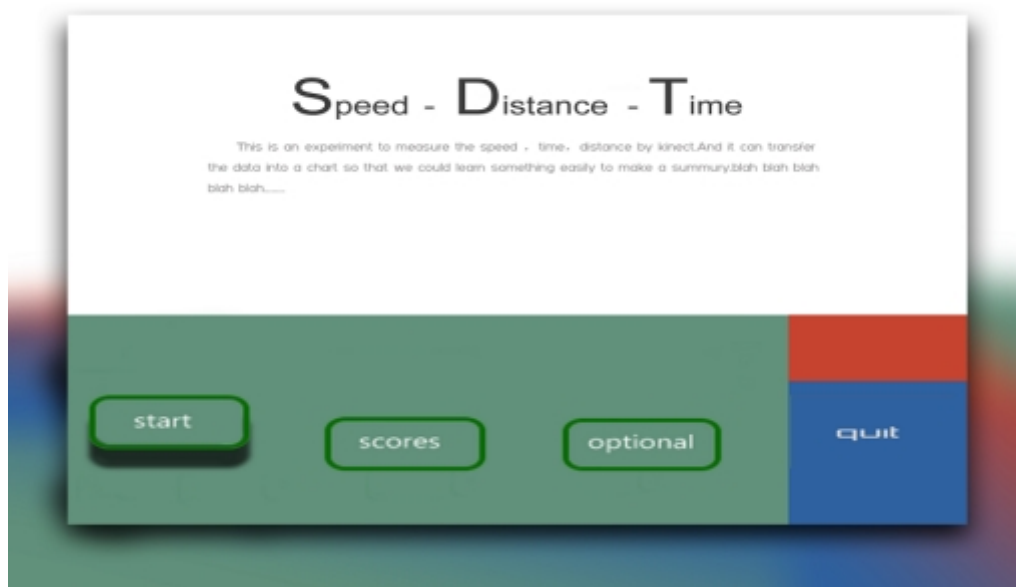
The main idea of the project was to develop a tool designed to assist physics class teachings in a high school classroom setting. This project is intended to make the basic physics principles easy to understand for students.

## Features

- The product provides a graphical representation to the values computed by kinect of time, distance and later on speed.
- The product is able to compare the simulations carried out by students in real time.

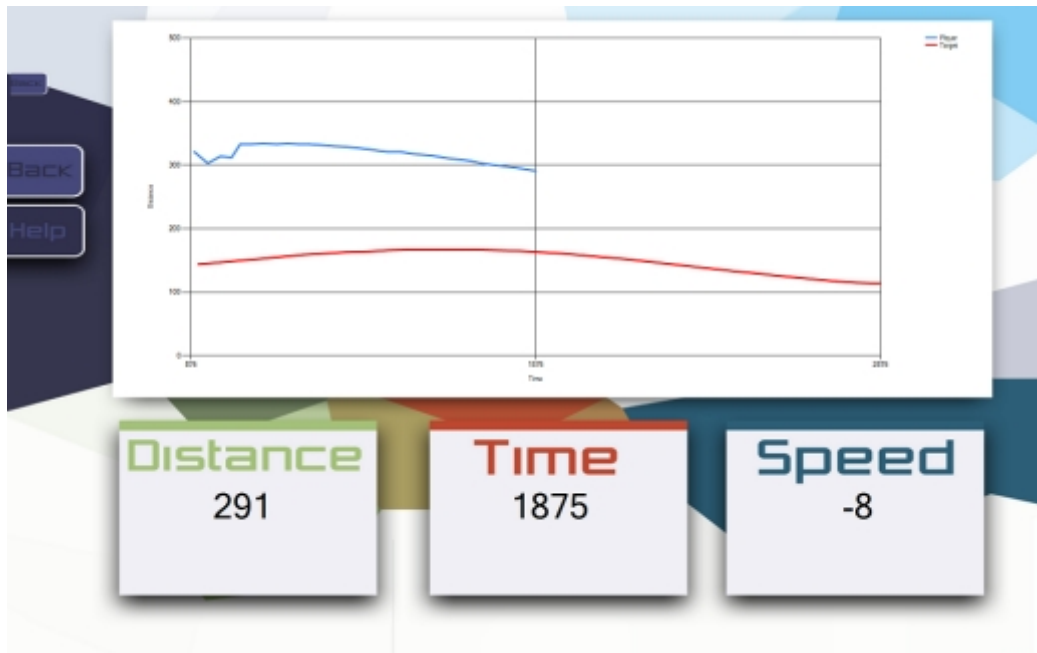
Along with these core requirements, there were following nonfunctional requirements:

- The product is capable of running the animations at a minimum of 10 frames per second (more the better).
- The product provides an interface to show statistics of carried out experiments.
- The graphs displayed to replicate animated to add more interest to the experiments.



Screenshot 1: Start screen





Screenshot 2: Game screen

## Organization and management

Our group originally consisted of nine members, four managers and five developers. One of the developers quit the project. The details of the group members are given below:

### Managers:

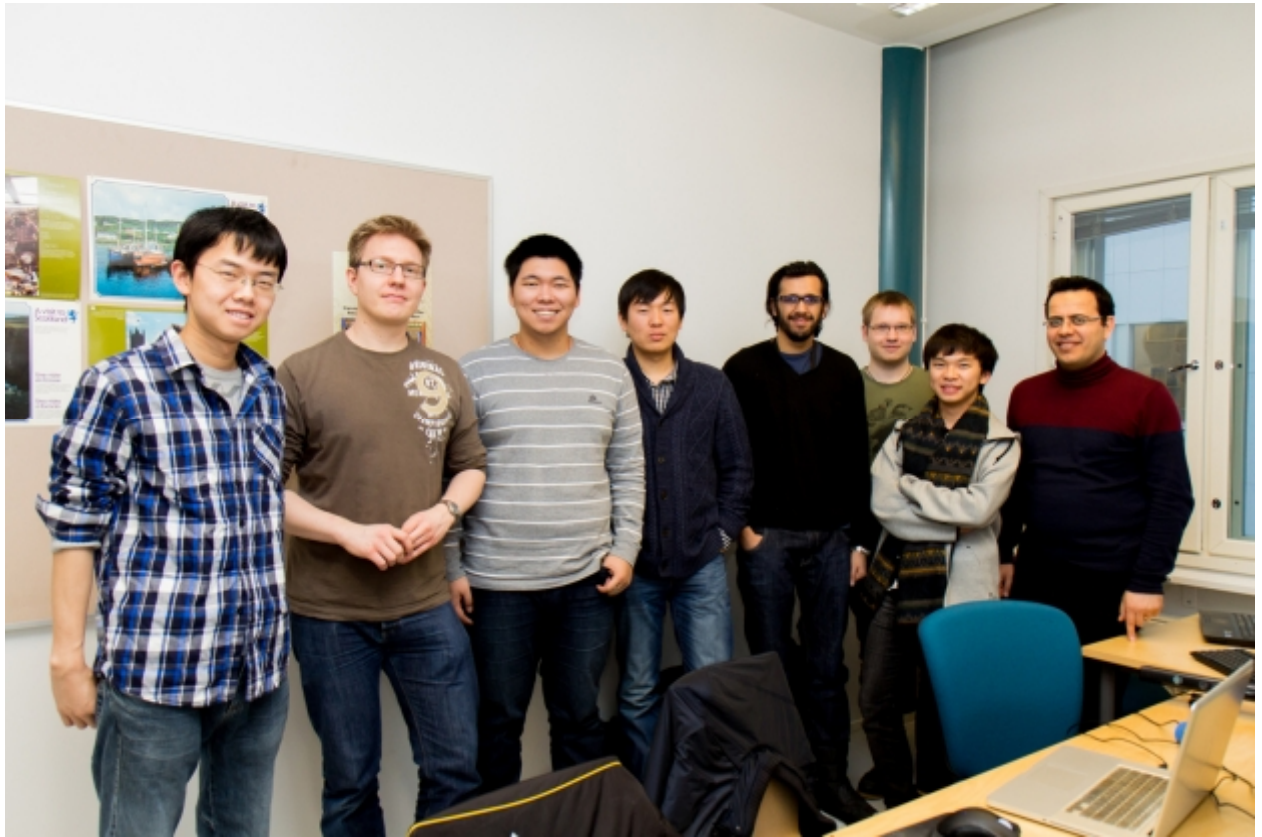
- Byambadorj Dulamsuren
- Reza Ahmadi
- Muhammad Farrukh Anwar
- Pengfei Lu

### Developers:

- Zhang Yi Chi
- Feilong Lin
- Aleks Lintuvuori
- Erik Veijola

Developers were divided into two groups, design and programming, overseen by corresponding managers respectively. Managers all had a specific roles each, such as scrum master, communication and coordination manager, documentation and reports management and tools management

## A picture of the team



*Team picture: (From left hand to right) - Pengfei, Aleks, Zhang, Byamba, Farrukh, Erik, Lin, Reza*

The team has three developers and two designers. In terms of skills, there was enough needed skills in the team. There was not a communication problem among the team members and we had much of communication during the project, from weekly meetings to chats in Facebook and via email.

The team members were mostly in the first year of their master studies. In the team there was people from different nationalities but there was not a big different in the style of working to make a big issue.

## Methods and tools

The development in this project was done with Microsoft Visual Studio 2012. We used C# as our programming language. For Kinect, we used Microsoft Kinect SDK. For user interface design, the designers used Adobe Photoshop software.

Project also used Redmine for tracking project progress, task assignment and working hours reporting. Redmine also contained our documentation and was invaluable for overall team coordination. Facebook served as the main communication hub for the team while Skype was used to arrange online meetings. SVN source code repository

was used and there was a benefit how SVN integrates to Redmine. Jenkins was used for continuous integration and building.

Most documents for the project were created with Google Docs, as this allowed the team members to edit and contribute simultaneously. For personal reports, other text editing tools, such as Microsoft Word were used.

## Project phases and development model

We used modified version of Scrum as a development model. Unfortunately it was not possible to organize daily scrum meeting obviously due to students' schedules and overall nature of the course. However, developers reported what they worked on, what were they going to do next and if they had any problems. Project sprints were three weeks long, except during the Christmas, as the members were not required to work during that time. The developers tested the software features locally as they finished them.

Task	Start Date	End Date	Status	Correspondence
Preliminary Report	20 September 2013	27th September 2013	Complete	Managers
Project Plan	28th September 2013	7th October 2013	Complete	Managers
Setup SVN	1st October	1st October	Complete	Managers
Architecture	7th October 2013	15th November 2013	Complete	Managers
Review #2	29th November 2013	29th November 2013	Complete	Managers
UI design	7th October 2013	10th November 2013	Complete	Developers
Review #3	12th December 2013	12th December 2013	Complete	Managers
Review #4	6th February 2014	6th February 2014	Complete	Managers
Implementation	7th October 2013	4th March 2013	Complete	Developers

Table 1: Project phases

## Experiences

Risk	Analysis
Keeping schedule	Between school work and study breaks, keeping a planned schedule is always a problem in student projects.  While the christmas break hindered our pace a little, overall we managed to keep with the course schedule by working hard at key moments especially at the end of the project.
Communication problems	Because of the fact that team members are not always working together

	all the time, problems could arise in areas such as code fixing and documentation.  Thankfully we have made full use of wide array of communication tools such as Skype and Facebook. We also arranged group work sessions to get everyone back at the same page at critical moments.
Previous experience	As is the case for any school project, students' experience level varies greatly in terms of skills. It is particularly highlighted when the project features a new technology.  This issue was handled by dividing the tasks among the group members according to their experience level in that specific area.

Table 2: Foreseen risks

Risk	Analysis
Team member quitting	A student quitting during the project puts a lot of pressure on the remaining team members. It is particularly hard when someone does it without any notice as was the case for us when one of our team member left the team.  We countered this issue by reorganizing the designers' team.

Table 3: Foreseen risks

Overall the team experience was mostly positive during the course, but there were some schedule related issues, which caused us to not reach the planned working hours estimation. Some functionality of the software had to be scratched due to this issue.

Working with Kinect was fun for us, particularly during the testing phase. Our team consisted of people of different nationalities and it was great working in a multi-cultural environment.

## What to do better next time

Since we had 4 managers in the team and 4 developers, next time the managers should participate in the development tasks to take the burden off of main developers. As half of the developers team had no experience in C# or working with Kinect, it put tremendous stress on the remaining team.

## Statistics

Team size	Dev. model	Start date	End date	Days	Hours
4+5	Scrum	September 17	March 5	170	1167.0

Table 4: General project information.

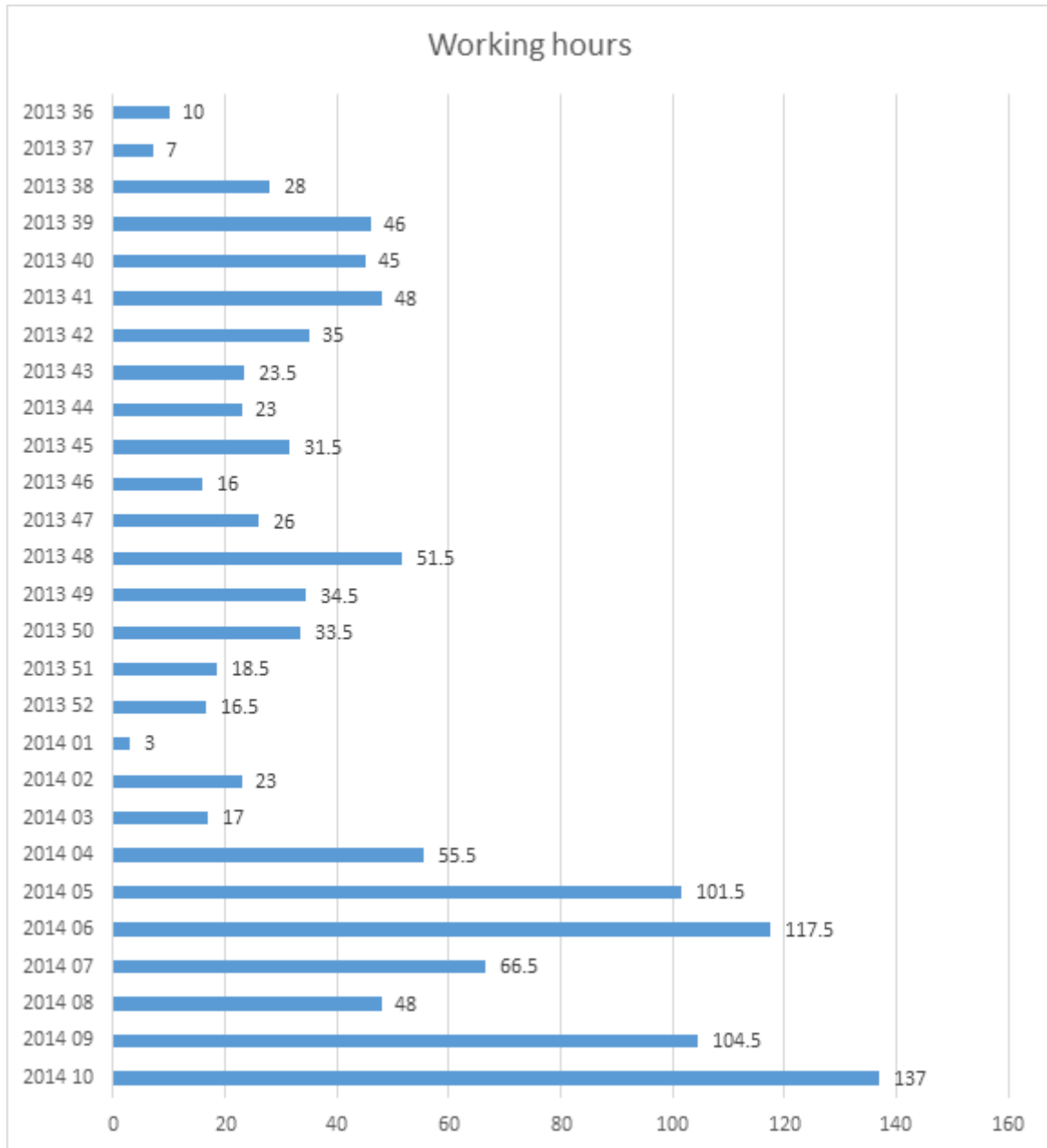


Diagram 1: Weekly working hours



Diagram 2: Group effort by activity

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
8	1	5	4	0	0

Table 5: Requirements and high-level design outcomes.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
3	3	0	0	2	1

Table 6: Design outcomes.

Document	Pages	Versions
Preliminary analysis	6	3
Project Plan	19	4
Requirements specification	1	7
Design plan	2	4
Test plan	10	2
Test report	16	2
Final report	9	3
Project's story	8	2
Weekly reports	24	24

Table 7: Documents.

Language	C#
LOC	830
SLOC	1102
Reused code	119
Reused and modified	100
Classes	10
Functions	112
Code revisions	71

Table 8: Code lines.

# Android Puzzle Game

## Overview

The scope of the project was to develop game with puzzles related to computer science, logical reasoning and/ or mathematics.

The main purpose of the game is to be a fun but educational way for people to get acquainted with the world of computer science.

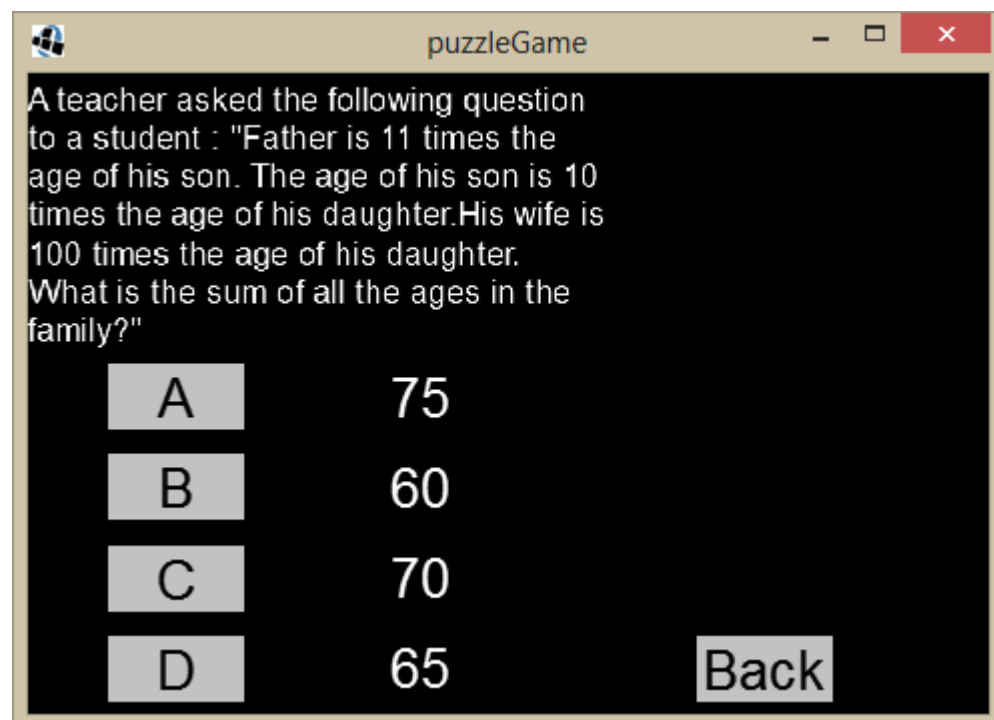
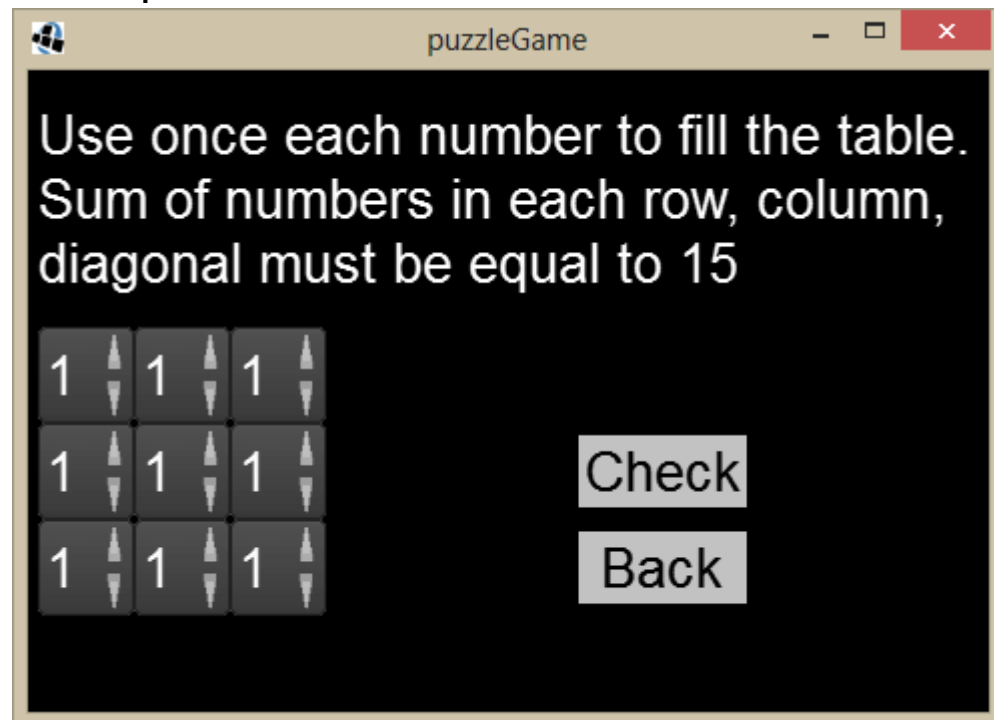
The game has three main GUI sections:

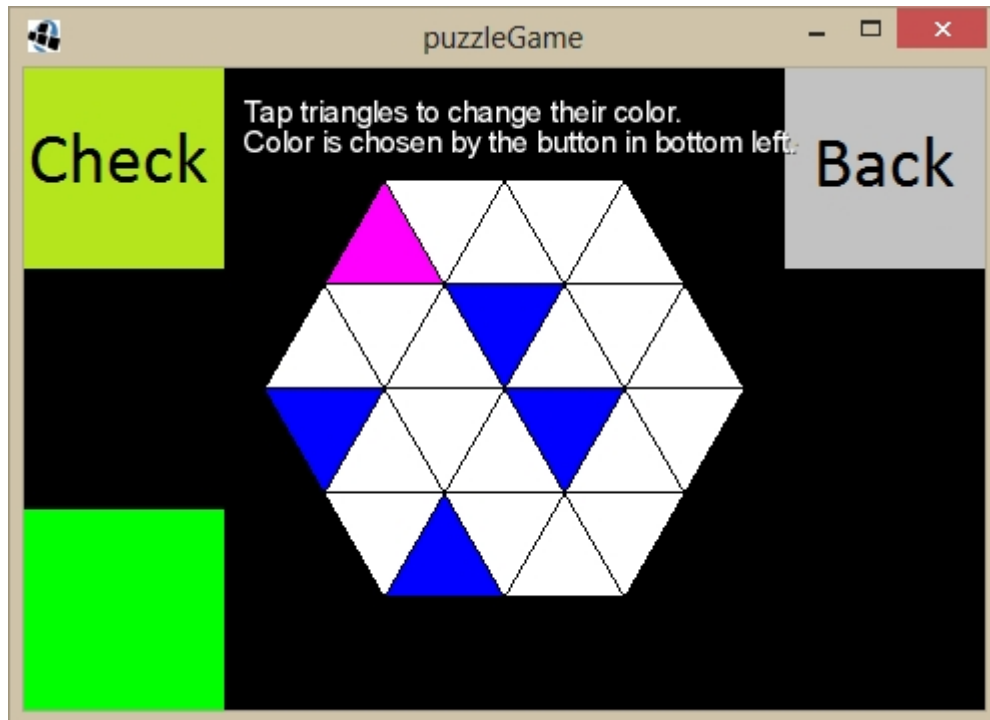
**The section selection screen (which map to enter) and map screen:**





Different puzzle screens:





### *Organization and management*

Juho Ylipoti	Manager Coding Manager
Eashan Salhotra	Manager
Ahmed Dawoud	Manager Testing
Jere Vahlman	Team Member
Mengyuan Yang	Team Member
Hang Do Minh	Team Member
Elena Betekhtina	Team Member



Picture: Part of the group during a workshop meeting. From right to left: Hang, Juho, Jere, Elena, Menguan.

### *Methods and tools*

Programming tools:

Android development tools (ADT Bundle)

Github (Version control).

is a web-based hosting service for software development projects that use the Git revision control system

Jenkins: is Continuous Integration framework

Sonar: platform for Continuous Inspection of [code quality](#).

Java

Libgdx Game framework: LibGDX is a game-development application framework written in Java with some C/C++ components for performance dependent code. It allows for the development of desktop and Android games by using the same code base

Planning, administrative and communication tools:

Redmine

Facebook group, Skype

Google drive

### *Project phases and development model*

- An incremental approach was adopted, iteration phases were: Plan >implement>Maintain. As more requirements arose, the system was be reviewed and updated.
- A model based testing approach was adopted. System functionalities were tested by the following steps: modelling>test case generation>adaptation>execution>analysis
- Tutorials sessions and workshops for team members for features implementation.
- Checkpoints and Milestones include Preliminary Analysis Meeting, Project plan, inspection Meeting, Review Meetings, Implementation of Puzzle Game with audio, and scoring system.
- Milestones were implemented in GitHub project repository

### *Experiences*

#### **Foreseen risks**

- Puzzle story not specified clearly.
- Project schedule and effort cannot be reliably estimated
- Multicultural group can be treated as a risk since communication problems may arise. To avoid it, we use project language as English and give instant updates to members via Facebook group
- Eashan, one of the project managers, might leave the project by 5<sup>th</sup> December 2013 as he is planning to graduate from his studies

#### **Risks not foreseen**

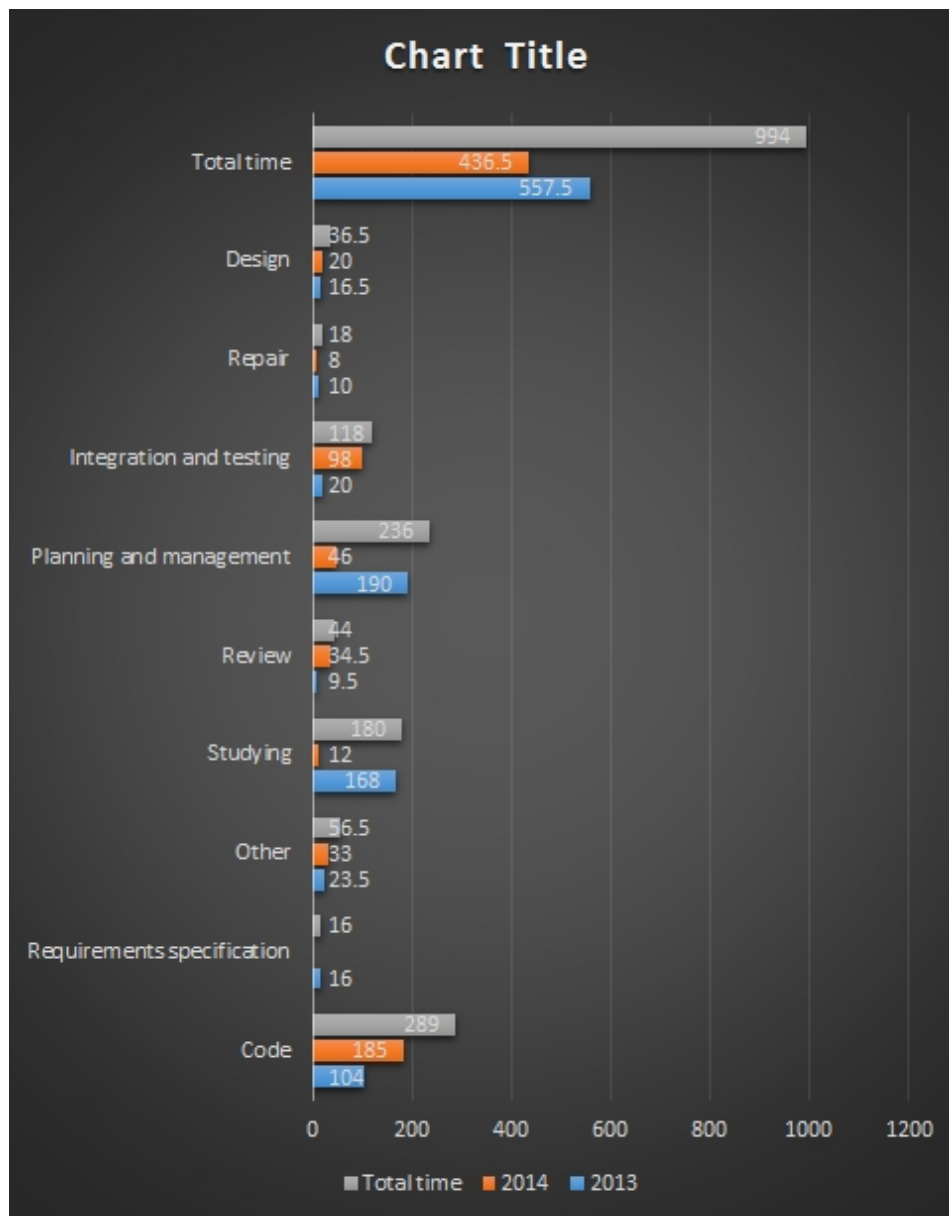
- Using Libgdx library was not compatible with testing framework.
- Troubles setting up Jenkins + sonar
- Issues using Git, mainly conflicts with class files

All in all, everybody views the project as a positive learning experience. It was a chance to get to participate in an interesting project utilizing fresh technologies.

### *Statistics*

Team size	Dev. model	Start date	End data	Days	Hours
3+4	Incremental Approach	09/24/2013	3/6/2014	164	~1000

Table 1: General project information.



Group effort by activity.

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
13			10	0	0

Table 3: Requirements and high-level design outcomes.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
		1			

Table 4: Design outcomes.

Document	Pages	Versions
Preliminary analysis	8	2
Project Plan	19	4
Usability analysis		
Requirements specification	1	1
Design plan		
User interface document	1	1
Test plan		
Test report	13	1
Usability test report		
Final report	15	1
Project's story	9	1
Weekly reports		

Table 5: Documents.

Language	JAVA
LOC	2930
SLOC	2930
Reused code	0
Reused and modified	0
Classes	24
Functions	184
Code revisions	153

Table 6: Codelines.

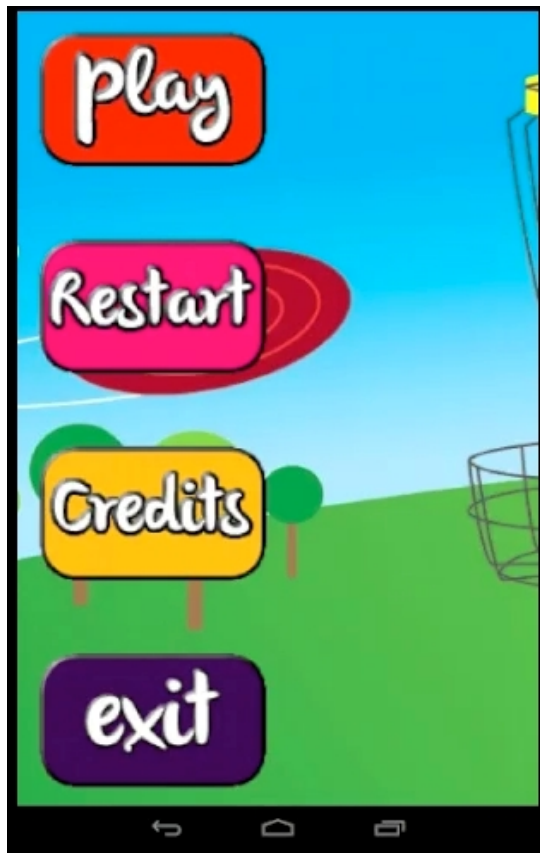
# Disc Golf Game Project

## Overview

The project created a 2D disc golf game for Android devices. Emphasis in the game is on a more realistic feel as opposed to many fantasy-themed disc golf games currently on the market.



Game view



Game menu

## Organization and management

Project managers:

Jarkko Huotari

Chenlu Wang

Chenyu Wei

Developers:

Lai Qinghui

Silvia Rubio

The project was originally assigned four developers but one cancelled her

participation for project work course. Then one member quitted from the project during the second sprint.



Disc golf project team.

Management was done in face-to-face meeting and through emails, Redmine wiki and Facebook.

## **Methods and tools**

The game is programmed in Java and it uses LibGDX game framework. LibGDX is very useful framework that simplifies Android development.

Coding is done using Eclipse with Android Development Tools plugin. Eclipse with ADT is a good tool for coding although it is a bit buggy. Android emulators that come with this tool are very slow and almost useless. Its a good practice to plug in your Android device when using Eclipse with ADT so you can run and debug the code in a real Android device.

Automated unit tests were written using jUnit and run on Jenkins server. SonarQube plugin in Jenkins was used to monitor coding metrics. Unit testing and coding metrics might have been more useful if we had had more time or a more experienced development team.

Redmine was used for project management and documentation. It seemed adequate for that.

Subversion was used for version control. It caused some problems with some svn clients not adding new created files into commits. Some other version control system might have been more user friendly.

Redmine, email and Facebook were used for written communication between project members, supervisor and client. In Facebook a group called “Disc Golf Project” were created for project-related communication.

Microsoft Word was used to edit deliverable documents. Microsoft Powerpoint was used to create project presentations. Microsoft Excel and Google Docs were used to create product and sprint backlogs.



## Project phases and development model

Development model used in the project was a modified version of Scrum. There were two major modifications made to Scrum development model for the purpose of this project. Firstly, daily Scrum meetings were replaced by weekly meeting because the project members' schedules do not allow daily participation for this project. Secondly, Scrum is created for software development teams working on a single office but this project has a dispersed team.

Use of Scrum in the project was problematic. Good use of Scrum would have required daily meetings and a lot more daily communication to keep the project progressing. Inexperienced developers could have benefitted from a more traditional and formal approach where everything is planned strictly beforehand. Also many members didn't seem to consider sprint deadlines as 'real' deadlines.

The project was divided into five phases: an initial planning phase and four sprints. A new playable version of the game was supposed to be produced in each sprint. However everything didn't go as planned. Game produced in first sprint wouldn't run on Android and wasn't really playable in Windows either. Game produced in second sprint ran on Android but wasn't playable. Game produced in third sprint was playable but still missing many basic requirements. Deadline for the fourth sprint had to be extended for three weeks but we finally produced a playable game with all the basic features agreed on with the client.

## Experiences

The project met almost all foreseen risks listed on project's project plan and test plan. We had communication problems caused by language difficulties and in some point lack of communication. The time ran out for the project and many features had to be cut because of that. Deviations from the design plan and the most active developer quitting during the project forced us to rewrite a large portion of the implementation. Implementation was not ready for testing in time throughout the project. Unfamiliarity with the technology used caused some problems. Members didn't work as much for the project as they had planned.

Lesson here is to take risk management more seriously. Prepare for all foreseeable risks and actively work for preventing them. Effects of many risks could have been mitigated by putting more effort into communication.

## Statistics

Team size	Dev. model	Start date	End date	Days	Hours
3+2	Scrum	11.9.2013	28.3.2014	200	932

*Table 1: General project information.*

Activity	Planning and management	Req. specification.	Design	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	381.5	24.2	48	212	35	21.5	14.5	152	43.3	<b>1430</b>
%	40.9%	2.6%	5.2%	22.7%	3.8%	2.3%	1.6%	16.3%	4.6%	100%

*Table 2: Group effort by activity.*

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
47	5	0	7	0	0

*Table 3: Requirements and high-level design outcomes.*

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
12	0	0	0	0	0

*Table 4: Design outcomes.*

Document	Pages	Versions
Preliminary analysis	6	1
Project Plan	25	7
Usability analysis	-	-
Requirements specification	5	1
Design plan	12	6
User interface document	-	-
Test plan	23	8
Test report	3	3
Usability test report	-	-
Final report	12	1
Project's story	5	1
Weekly reports	26	1

*Table 5: Documents.*

Language	JAVA
LOC	1582
SLOC	1582
Reused code	0
Reused and modified	0
Classes	18
Functions	110
Code revisions	202

*Table 6: Codelines.*

# Seek N Share

## Overview

Seek'n'Share is a mobile learning environment where students can wander around and use tablet devices to complete given assignments. Students can also use software on tablet devices to capture video, audio and/or photos and compose a presentation from them. Composed presentations are stored to a backend system.

The scope of the project is to produce an editor that will enable users to edit presentations after they are stored to backend database, like reordering of objects on presentation, adding new objects to a presentation, merging of presentations, removing objects from presentation, Playing a presentation, converting presentation to other formats (eg. video or Youtube), easy localisation, editor must be usable with desktop PC, editor has to be built on top of web technologies. Editor will also be able to convert presentations to other formats for wider consumption.

## Organization and management

Person	Role	Job
Antti Varjoinen	Project Manager	Project Integration and Scope management
Jiadong Liu	Project Manager	Project Integration and Scope management, Risk management
Renfei Zhou	Project Manager	Cost, Quality management and Testing
Thrushna Nalam	Project Manager	Communication, Human resources, Scope management.
Jie Chen	Developer	Coding, Integration
Biruk Yemane	Developer	Coding, Integration
Faiza Ahmed	Developer	Integration, Debug, Testing
Dan Yang	Developer	Coding, UI Interface

## Methods and tools

Communication:

Facebook:

Function: To communicate with team members for all latest updates, questions...

Level of Useful: Low.(lack of active facebook users)

Skype

Function: Online communication among the team members.

Level of Useful: Low. (Lack of time management)

Email

Function: Keep communication in the team by email, especially for weekly report.

Level of Useful: High.

SVN

Function: Upload and download the document of code.

Level of Useful: Medium. (Lack of usability knowledge)

Tools used in the project are listed below:

Tool	Description	Link
Web browsers	The application is also manually tested with decent web browser like IE 9, Firefox, opera and Chrome to ensure correct user interface and usability	-
AngularJS	Javascript framework that the project builds on. Ng-boilerplate is used to make development easier and faster	<a href="#">AngularJS Ng-boilerplate</a>
Code editor	Any decent text editor but code editors like Brackets are recommended	<a href="#">Brackets</a>
Node.js	Back-end javascript environment supporting development	<a href="#">Node.js</a>
Karma	Test runner for AngularJS. Used in the project to do unit testing and end-to-end testing	<a href="#">Karma</a>
NPM	Package manager for node.js	<a href="#">NPM</a>
Bower	Dependency handler for the project	<a href="#">Bower</a>
Grunt	Task runner supporting development process	<a href="#">Grunt</a>
Trello	Scrum board and issue tracker for the project	<a href="#">Trello Project Scrum board</a>
Dokuwiki	Collaboration and documentation platform for the project	<a href="#">Dokuwiki</a>

		<a href="#">Project's wiki</a>
Jenkins	Continuous integration server used for QA	<a href="#">Jenkins Project's Jenkins job</a>

[http://jenkins.sis.uta.fi/jenkins/job/WWW Editor for Seek n Share/](http://jenkins.sis.uta.fi/jenkins/job/WWW_Editor_for_Seek_n_Share/)

## Project phases and development model

The development model was a modified Scrum i.e. Trello. acting as the scrum board. Lack of commitment caused obstruction in efficiently following the development model.

The team members did not actively participate in the scrum meetings in the early stages of the project. We could not follow the schedule of planned sprints.

In the initial schedule of the project, 7 sprints were planned. Besides, there were milestones as recommended by the course guidelines i.e. Project plan inspection meeting, three review meetings and a final meeting. In the actual we were all able to complete only 4 sprints. The fate of the project was decided after discussions in the emergency meeting.

Below table describes the checkpoints and milestones of the project.

Task	Project Initial Schedule		Project Actual Schedule		
	Start Date	End Date	Start Date	End Date	Progress
Preliminary analysis	22.09.2013	30.09.2013	22.09.2013	30.09.2013	Completed
Project Plan	30.09.2013	08.10.2013	30.9.2013	25.10.2013	Completed
Sprint I	22.10.2013	18.11.2013	22.10.2013	19.11.2013	Completed
Sprint II	19.11.2013	01.12.2013	19.11.2013	2.12.2013	Completed
Sprint III	02.12.2013	17.12.2013	3.12.2013		Incomplete
Sprint IV	17.12.2013	14.01.2014	3.12.2013		Incomplete
Sprint V	15.01.2014	28.01.2014	No Progress		
Sprint VI	28.01.2014	11.02.2014			

Sprint VII	11.02.2014	25.02.2014			
First Review meeting			18.11.2013		Completed
Second Review meeting			17.12.2013		Completed
Third Review meeting			Emergency meeting- 20.2.2014		Completed
Final meeting			Week no 12		

## Experiences

### Foreseen risks

- Team members are not so experienced developers or coders.
- Some group members have got jobs and some with kids, so it is hard for organizing some meetings with everyone there.

### Risks not foreseen

At first when we decide on the interface we will use, we chose JSON. However, JSON is totally new to our developers and they will have to spent quantity hours on learning. It's too late while we realized it does not suit our project.

Describe here your experiences from the project and what you can do better next time! Share your experiences with other groups and readers of this document.

- Working together is always better than working individually. Group members can discuss and solve problems efficiently in time.
- It is extremely important to choose a proper framework or technique for the project. Not until the final stage of the project did we find that what we choose for the project is too difficult for us.
- Schedule in advance would never be wrong. Since we are working in a team, organize group meetings with everyone participate would be important.
- Tasks are better to be assigned with group member's ability and interest.

## Statistics

Team size	Dev. model	Start date	End data	Days	Hours
4+4	SCRUM	14/09/2013	15/03/2014	183	1244.5

*Table 1: General project information*

Activities	Sep-28	Oct-26	Nov-30	Dec-28	Jan-25	Feb-22	Mar-29	Total
Planning and management	60.5	78	108.5	110	19.5	32. 5	32	423.5
Code	0.5	0	30.5	17	37	13	20	98
Design	0	0	26	3	11	7	2	49
Studying	35.5	58.5	84	49	46	36	3	312
Other	16.5	38.5	41	8	10	22.5	4	140.5
Integration and testing	0	0	3	3	0	8	0	14
Review	0	1	5	5	1	3	5	21
Requirements specification	1	3	0	10	0	1	0	15
Repair	0	0	0	0	0	2	0	2
Total	114	234	271	177	115.5	252	81	<b>1244.5</b>

*Table 2: Group effort by activity*

Number of requirements	Use-cases	UI screens	Database diagrams	Database tables
33	NO	NO	NO	NO

*Table 3: Requirements and high-level design outcomes*

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
/	/	/	/	/	/

*Table 4: Design outcomes*

Document	Pages	Versions
Preliminary analysis	No	No
Project Plan	21	1.5
Usability analysis	No	No
Requirements specification	In trello	In trello
User interface document	No	No
Test cases	8	1.0
Test report	6	1.0
Usability test report	No	No
Final report	3	1.0
Project's story	5	1.0
Weekly reports	19	/

*Table 5: Documents.*

Language	JSON
LOC	780
SLOC	/
Reused code	0
Reused and modified	0
Classes	/
Functions	Hard to Calculate
Code revisions	Beta

*Table 6: Codelines*



# Flowing Hours

## Overview

Purpose of the project was to create a web application named Flowing Hours, a Time Management Tool that enhances both productivity and well-being. The idea of the application is to help people manage their time, both professionally and personally. To quote our client:

“For many people work life today is demanding, stressful and even exhausting. We have to be productive and effective all the time but still manage to maintain a certain level of wellbeing. These two things can be very difficult to combine. On the one hand we have a lot of different time and project management tools, but these cause many people more stress instead of reducing it. On the other hand we have all kinds of Slow Life philosophers who say that we should work less in order to improve our wellbeing.”

At the beginning of the project the aim was to produce a first version of the application for the client to use for managing her time, and to use as a starting point in designing the application further. Due to complications during the development, it was decided however to change the end result of the project to be a demo of the actual application (see screenshots below) and documentation including a list of its features (designed together with the client but not necessarily implemented into the demo), a technical solution design (e.g. what kind of techniques and programming languages to use) and a test plan.

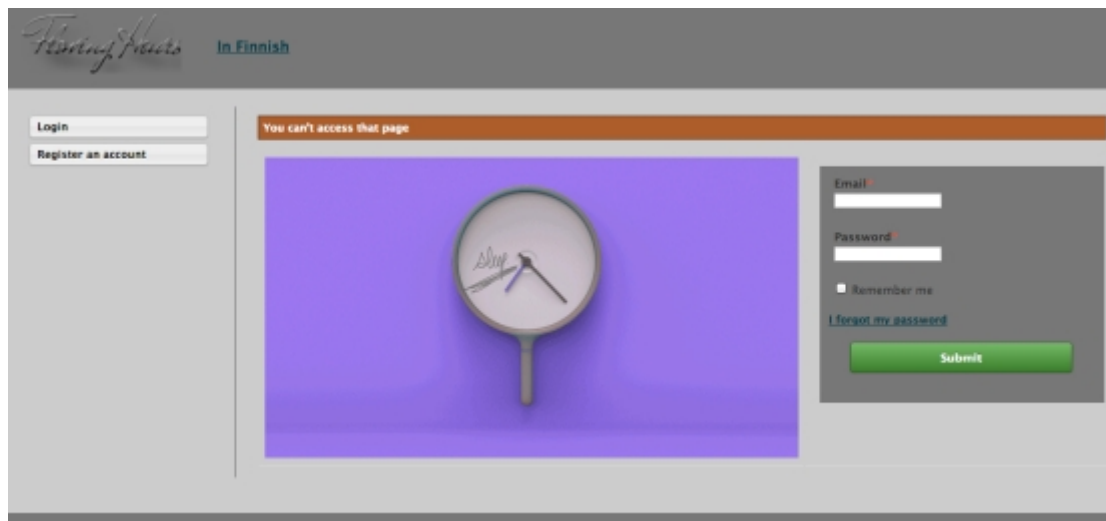


Image 1. Screenshot of the front page of the application.

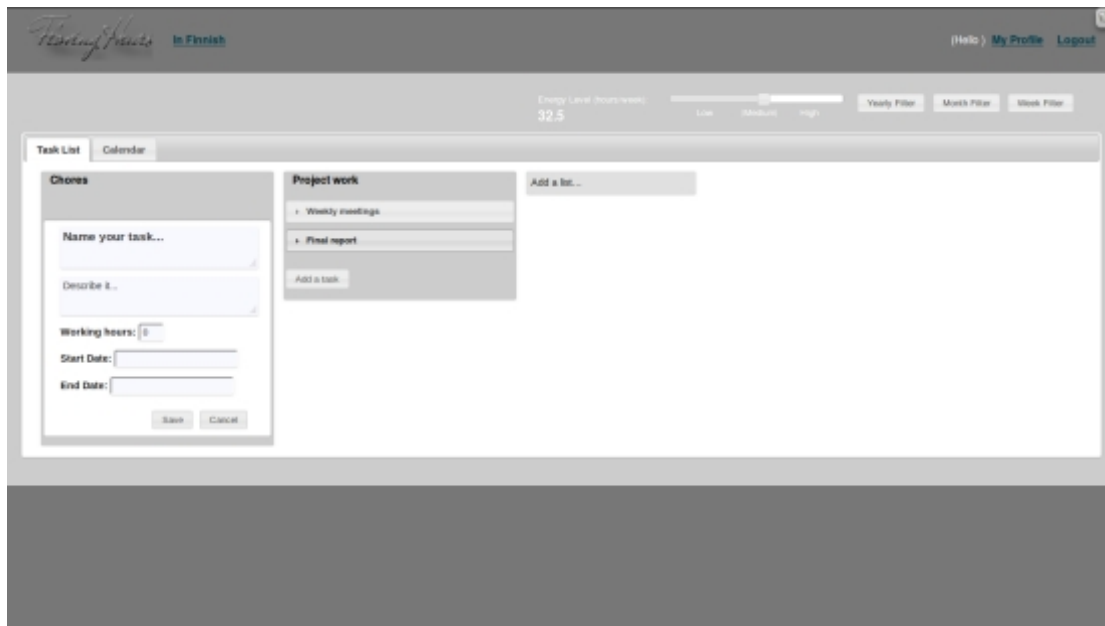


Image 2. Screenshot of task lists.

## Documentation

Documentation delivered at the end of the project included:

- Technical documentation
- System specification
- Project plan
- Project story
- Test plan

## Demo of the application

The final demo of the application included the following features:

- User login/logout
- Admin control panel
- Tasks
- Task lists (both Task lists and Time task lists)
- Calendar
- Feedback system
- Energy level settings
- Ability to choose calendar wallpaper

- Color schemes
- UI in English & Finnish
- User profile

## **Organization and management**

### **Managers**

Manager Juha Kaura is a fifth year student at the University of Tampere and is currently working on his Master's degree in Software Development. His programming experience extends mainly to web applications and languages like PHP, Java, Groovy, Python, Javascript (and many frameworks, techniques, architectures etc). Juha is currently working as a part time web developer for Sofia Digital Oy. Juha's main responsibilities included giving technical advice, making technical decisions, and general team management.

Manager Siiri Tammisto did her Bachelor's degree in computer sciences in TAMK and graduated in 2008. In 2011 she started her Master's degree on Information Systems in Organizations in the University of Tampere. She is currently working on her thesis and working part time at Liaison Technologies. She has some experience in graphic and web design, html and css, but doesn't have any real programming skills. Due to her lack in programming experience, she focused on managing documentation and scheduling of the project. She also spent time with the client discussing the future requirements and features of the application.

Manager Ao Li took his Bachelor's degree in Software Engineering in Wuhan University in China this year. Now, He is a degree student in SIS on Software Development. He has some experience in C#, JAVA and know a little about Web fundamental language, but don't have experience in web design. During this project Ao's main responsibilities included team management and testing.

### **Developers**

Developer Muhammad Qasim is from Pakistan and did his Bachelor's degree in Computer Science in his home country. Now, he is doing his Master's degree in Software Development in the University of Tampere. He has programming experience of C#, Java and VB.NET and also some knowledge of web development languages such as HTML, CSS, and JavaScript. Muhammad was responsible for the application's feedback features.

Developer Abrar Aamer is also from Pakistan and also did his Bachelor's degree in Software Engineering there. He is now doing his Master's degree in Interactive Technology at the University of Tampere and is currently working on his thesis. He describes himself as an average programmer in PHP, HTML, JavaScript but he has worked on custom PHP and Joomla projects. Abrar is working at the moment as a freelancer web developer. Having the most programming experience of all

developers, Abrar's responsibility was the backend implementation in the beginning of the project. As the project went on, Abrar's responsibilities grew to cover the frontend implementation as well.

Developer Wen Hao is an exchange student from China. She is familiar with programming languages like C, C++ and Java but has only little real programming experience. She has done some UI designing for an Android mobile application and is now hoping to learn about UI designing for web applications. Her responsibilities involved the UI design for the application as well as implementing it by writing CSS.

The original project team had one more developer, but for some reason or another, he never got fully involved with the project and after not replying any e-mails or showing up to team meetings, he is considered a dropout.

### **Client**

Inna Thil is a Manifestation Mentor and a businesswoman from Tampere, Finland. Inna suffered from depression and burnout, but while fighting to get through her difficulties, she started to focus more deeply on what she really wanted to do with her life, and finally found happiness. She now helps other people suffering from anxiety, depression or unhappiness to find their own, true path in life.

For more information about Inna go to <http://www.omanelamaniluoja.fi/>.

## **Methods and tools**

### **Development tools**

Coding/text editor: NetBeans IDE 7.2.1

Version control: GIT with NetBeans IDE 7.2

FTP: Filezilla 3.7

Web server: Apache 2.2

Database: MySQL 5.5

Server-side language: PHP 5.3

Web Server Environment: WAMP and XAMPP both

### **Management tools**

Trello was used for appointing and maintaining team tasks and individual tasks. We used Google Docs for hosting all project documentation and working hours (a spreadsheet). Additionally, we used Flowdock for instant messaging but it did not gain popularity among the developers. Flowdock would have been quite useful for simultaneous development due to its easy-to-use interface and versatility. Trello and Google Docs were used actively throughout the project and turned out to be crucial for team management.

## **Project phases and development model**

When evaluating whether the development model and methods used were working properly or not, or if they would meet the course's criteria, there is one very important fact to remember: since in the beginning of the project there were no real application feature specification, only an idea of an application. Including the developers to join the work right from the beginning was very difficult. Sure, there were a few features known right from the beginning that would be needed for the application. But to manage and develop a project efficiently, more 'background work' should be done before moving on to the actual implementation or developing phase.

This said, with the knowledge of what we had in the beginning of the project, we chose a flexible development model that allowed us to work individually and independently. We avoided using Scrum for its strict nature of roles, meetings etc. To help managing the schedule of the project, the project was divided into four phases: Start, Development, Testing and Wrap-up. Not unlike to the Waterfall method, the team could move into the next phase only after the latter would be (at least almost) completed. However, in the end of the project the Testing-phase was non-existing since the development of the demo of the application continued right to the last week of the project.

In the beginning, we decided that instead of sprints we would slice the workload into features that need to be implemented. Then, the features were put in order with the least significant being last to be implemented. Once a feature was implemented, we moved onto another feature.

Unlike in the waterfall model, we did not have predefined processes for coding or testing where one could only revert back to previous process. Instead, we tested a feature when it was implemented and then fixed the bugs and moved on. This way, we reacted to changes and other things just in time.

One could think of this model as an organic software development model where specification design, project planning, coding and testing is a continuous process and the team members react to changes when there's a specific need for it.

We had a team meeting once a week throughout the lifetime of the project and for each meeting there were specific tasks to be completed. As such, our development model reminded that of Scrum where the sprints define the basic unit of development and there are meetings for planning the sprint and meetings for evaluating the outcomes of the sprint. In each meeting we set up tasks for the next week and inspected the results of the last week's work.

The following table describes the major deadlines of the project.

<b>Name</b>	<b>Date</b>	<b>Responsibility</b>	<b>Description</b>
First team meeting	19.9.2013	Project team	introduction of the whole group
First meeting with the client	23.9.2013	Project team + client	getting to know the client and the product
Preliminary analysis	24.9.2013	Managers	preliminary analysis of the project
Project plan inspection	15.10.2013	All stakeholders	project plan inspection
Project review + proto I	19.11.2013	All stakeholders	first proto of the application
Project presentation	4.12.2013	Project team	project presentation for other project teams
Project review + proto II	13.12.2013	All stakeholders	second proto of the application
Project review + proto III	30.1.2014	All stakeholders	final proto of the application
Final project presentation	5.3.2014	Project team	project presentation for other project teams
Final meeting	13.3.2014	All stakeholders	final meeting, delivery of all project deliverables to the client

Table 1. Deadlines of the project.

The following table lists the documents produced during the project.

<b>Name</b>	<b>Number of pages/slides</b>	<b>Number of revisions</b>
Preliminary analysis	8	1
Project plan	20	7
System specification	29	4
Technical documentation	9	3
Project story	11	1
Final report	20	3
Weekly reports	24 reports	-
Project presentation slideshows (initial, midterm and final)	6, 18 & 14	-
Review I, II and III slideshows	14, 10 & 11	-

Table 2. Documents of the project.

## Experiences

In the beginning, there were few risks that we thought were likely to come true. The requirements for the application we had seemed reasonable and we had at least one experienced web developer. We noted that the Google Calendar synchronisation could be difficult and the feedback system could prove to be challenging to implement. Of course, we acknowledged that already one of our developers, Miika, was not participating the course anymore. All in all, the workload did not seem overwhelming.

There were no crucial software related errors that could have stalled the development. As the project continued, it became clear that the team had difficulties with motivation, commitment, and technical experience. This was something that the managers did not anticipate and as such was an unforeseen risk to the success of the project. When the managers acknowledged this risk they made efforts e.g. set deadlines, email notifications etc. to ensure that tasks were done on time. This was not enough to correct the course and this eventually lead to more problems. One of the first group tasks were individual introductions so that the managers could have an understanding of each team member's capabilities. Therefore, the managers knew what to expect from the developers although this was an insufficient action to inspect the available team resources. For example, Muhammad's and Hao's level of experience surprised the managers when the implementation began: using Git, Netbeans, or writing CSS was more difficult than what the managers had expected. This could have been avoided by surveying the team resources more closely.

Furthermore, English as the language of choice for group discussion affected our communication as far as the requirements engineering was concerned. We quickly noticed that speaking with our client in Finnish produced better results and resulted in clearer understanding of the client's vision of the application. If everyone could have attended all of the meetings with our client the developers might have had a more concrete understanding over the requirements for the application and this could have lead to more motivated working efforts.

Regarding the development process, the chosen development model was unsuitable for the team. This was one of the major errors that the managers made and it cost the success of the project. In retrospect, the managers should have chosen a more process driven approach to the software development but naturally this would have meant a more dedicated, active, and contact-based interaction on the managers' behalf: schedule a coding session at least twice a week, supervise the development face-to-face etc. In reality, the managers' resources did not suffice for this kind of development process.

All problems mentioned above resulted in changing the objective of the project in end of 2013: the application implemented would only be a demo of the application, and more emphasis would be put to documentation and designing of future features.

## Statistics

Team size	Dev. model	Start date	End date	Days	Hours
3+3	Ad hoc	19.9.2013	13.3.2014	110	810

*Table 3: General project information.*

Activ ity	Plann ing and mana geme nt	Req. speci ficati on.	De- sign	Code	Integ ration and testin g	Revie ws	Repai r	Study	Other	Total
Hour s	305,5	51,5	23	232	40,75	2	9	119,5	26,5	810
%	37,72	6,36	2,84	28,64	5,0	0,25	1,11	14,75	3,3	100%

*Table 4: Group effort by activity.*

Number of requirement s	Pages	Use-cases	UI screens	Database diagrams	Database tables
37	38	--	12	1	6

*Table 5: Requirements and high-level design outcomes.*

Name	Number of pages/slides	Number of revisions
Preliminary analysis	8	1
Project plan	20	7
System specification	29	4
Technical documentation	9	3
Project story	11	1
Final report	20	3
Weekly reports	24 reports	-
Project presentation slideshows (initial, midterm and final)	6, 18 & 14	-
Review I, II and III slideshows	14, 10 & 11	-

*Table 6: Documents.*

Language	PHP, CSS, JavaScript, HTML, SQL
LOC	373508
SLOC	4767
Reused code	--
Reused and modified	--
Classes	--
Functions	--
Code revisions	78

*Table 7: Codelines.*

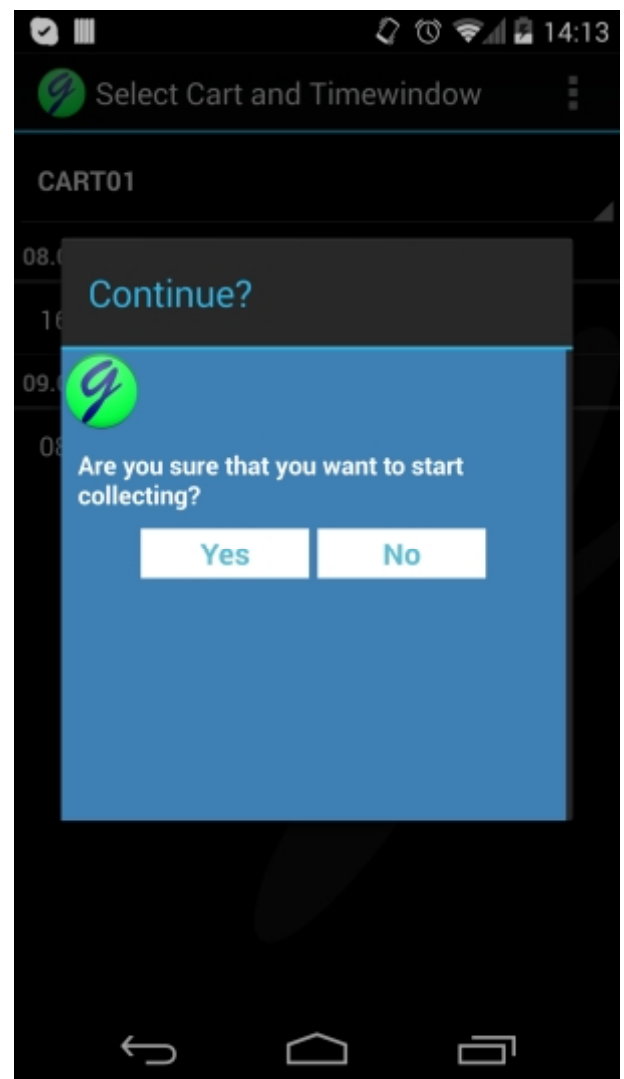
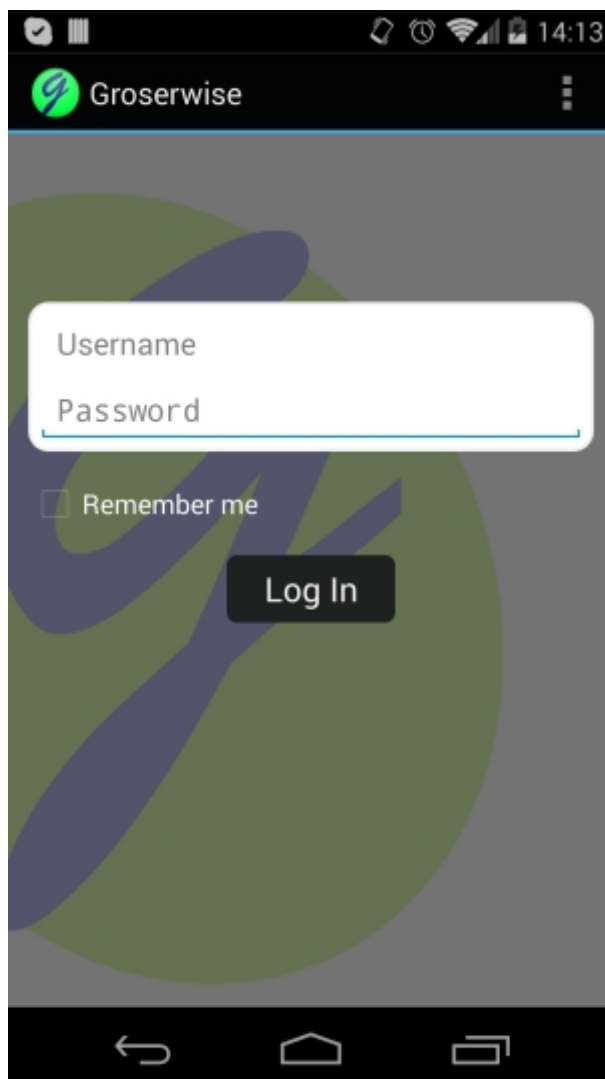


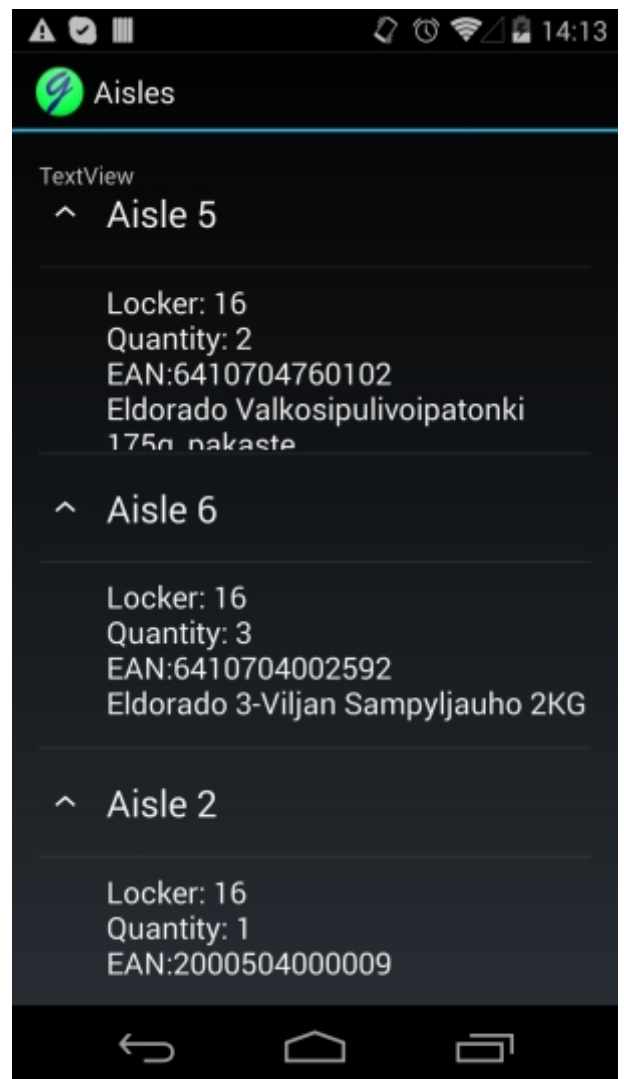
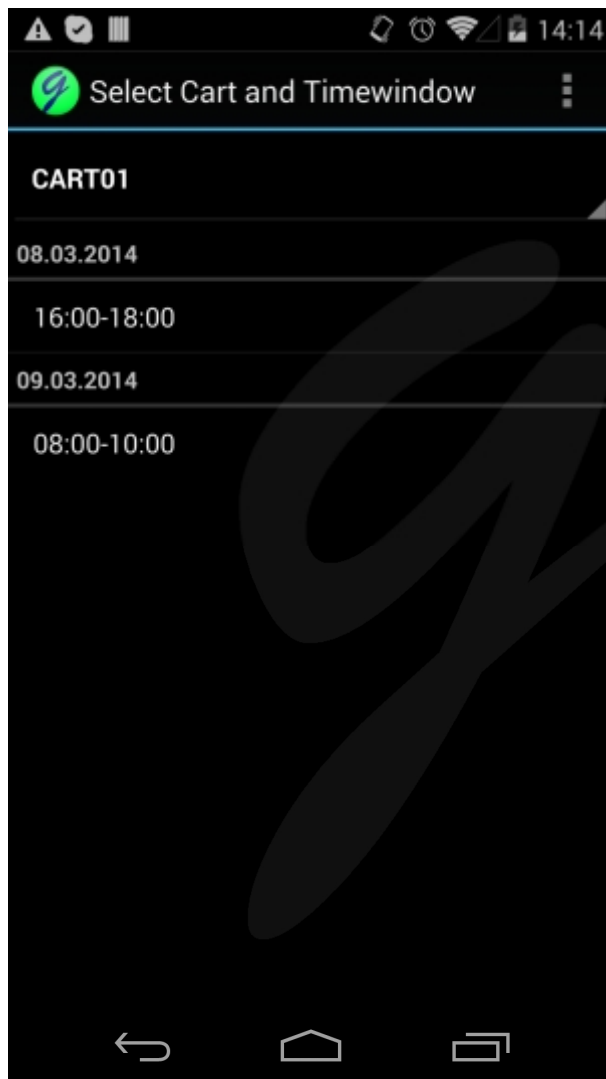
# GroSerwise Systems

## Overview

The system's purpose is to help grocery stores to sell products online. Part of the system, developed within this project is an Android program for collecting items, as well as an API program on server side. The API program stores everything to the database and calculates everything. Then it gives all information to other programs.

The idea came from the notion that grocery stores were not in e-commerce business. Reason for this was too big expenses for collecting and delivery with methods used today. There is a need for a system that cuts those expenses and the system developed within this project is expected to meet this need.





## Organization and management

Our project group consisted of 4 managers and 3 developers. In the following is a list of group members and roles.

### Project Managers:

Golnaz Sabetnejad (writing documents, doing weekly reports, arranging stuff), Ma Xiaolan (weekly reports, editing documents, arranging stuff), Otso Leppälä (scrum master, messaging with client, arranging meetings), Iulia Adomnita (editing documents, arranging meetings, administrating Jenkins, SVN, Google community).

### Developers:

Mikko Liukkonen (design and implement API and database), Xu Menglin (UI design, helping with the collecting program implementation), Rahul Arora (designing and implementation the Android application).

## **Methods and tools**

We used Eclipse IDE with Android SDK and Android emulator and testing devices for development and SVN for version control. GroSerwise System company gave us two testing devices to use in our project. For testing we used Android SDK:s integrated testing framework. For documentation we have used javadoc.

The developing team has reached a consensus on applying and following the guidelines and principles declared by Google regarding Android development.

As we used object-oriented programming language and environment, we did all required modelling with UML notation.

Redmine of Tampere University was used as a project management software. Redmine was used to track working hours, assign tasks to team members and get statistics about project's progress.

Google drive was used to share knowledge about technological issues, writing and storing formal documents, such as Preliminary Analysis and Project Plan, weekly meetings' minutes, questions about the project and weekly reports.

For version control we used university unit's Subversion. Everybody got a permission to use Subversion at early phases.

As communication methods we utilized Skype and Google +. Since weekly meeting was not enough to discuss about all issues related to the project, we had Skype meetings between managers and developers during the course. It was also used to distant consulting between managers and developers. Email was used to send official informations and urgent questions.

## **Project phases and development model**

### **Development model**

Development model for this project was modified Scrum. As a student project it is impossible to use "pure" Scrum. Project team had weekly meetings but not daily Scrum meetings. Project schedule was divided into phases and phases further into shorter sprints, but sprints were actually only written in schedule but not supervised as the management concentrated on phases. Review meetings with project supervisor and the client were scheduled in the ends of development phases.

Since the scope of the project was quite big, we prepared the requirement document for writing and collecting requirements. In each review meeting, we went through the requirements and tried to prioritize the requirements with team members, supervisor and the client based on our skills and time schedule.

At the very beginning of the project a lot of time was be used to studying and learning new programming environments and there were not sprints at all in the beginning. We did a lot of demos and meetings with the client as we didn't know if and how all requirements could be achieved.

### **Project's phases:**

#### **Sprint 1:**

11.09.13: The team received the project proposal;

13.09.13 First meeting with the team;

17.09.13: First meeting with the client. Requirements document started;

17.09.13 - 28.02.14: Requirements elicitation and management. Maintaining the requirement document up to date;

25.09.13: Meeting with the client and Pekka;

25.09.13 - 04.10.13: Get server and install GlassFish;

25.09.13-13.12.13: Database design and implementation;

#### **Sprint 2:**

17.09.13 - 28.02.14: Requirements elicitation and management. Maintaining the requirement document up to date;

25.09.13 - 04.10.13: Get server and install GlassFish;

25.09.13-13.12.13: Database design and implementation;

10.10.13: Preliminary Analysis meeting;

21.10.2013 - 15.11.13: UI design for Android programs;

#### **Sprint 3:**

17.09.13 - 28.02.14: Requirements elicitation and management. Maintaining the requirement document up to date;

25.09.13-13.12.13: Database design and implementation;

21.10.2013 - 15.11.13: UI design for Android programs;

10.11.2013 - 20.01.2014: Show collecting route in the collecting program;

01.11.13: Project Plan meeting. Project Plan document written;

01.11.13 - 07.11.13: First personal report submitted.

20.11.13: Requirements meeting with the client and Pekka;

#### **Sprint 4:**

17.09.13 - 28.02.14: Requirements elicitation and management. Maintaining the requirement document up to date;

25.09.13-13.12.13: Database design and implementation;

10.11.2013 - 20.01.2014: Show collecting route in the collecting program;

01.12.2013 - 04.12.2013: Project presentation;

10.12.13: 1st Review meeting;  
01.01.2014 - 20.01.14: Send collecting information to API;  
01.01.14 - 14.01.14: Second personal report submitted.

### **Sprint 5:**

17.09.13 - 28.02.14: Requirements elicitation and management. Maintaining the requirement document up to date;  
14.02.14: 2nd Review meeting;  
28.02.14: 3rd Review meeting;  
5.03.14: Final Presentation;  
12.03.14 - Final & Feedback meeting;  
10.03.14 - ? Third personal report;

## **Experiences**

Like everything in life gives us some experiences, this project also gave us quite many of them - life related lessons, as well as career related ones. All team members got experience in multicultural communication, we learned to use tools and got hands-on experience on managing and developing a real-world application.

### **What to do better next time**

Our project can be implemented better next time if we take these factors into consideration and do as follows:

Make a workable project plan and keep it up to date. Making an effective project scheme and update the status once a week help us know the progress of the project better, and decide whether to make changes accordingly.

Make a workable schedule and be as stricter as you can with deadlines. Unlike project plan, schedule should include more detailed tasks with strict deadlines. It needs to be reviewed and adjusted until the application is delivered to the client. If we stick to the schedule, the project will be more effective.

Establish a good contact with the client and all team members. We're faced with the communication problems because our client is a Finnish speaker while only two of team members are fluent in Finnish, so most of the communication are held between them.

Also the documents need to be translated to English manually before shown to other members. Thus, Otso and Mikko have to work more than others. It might be better if we list the import points of client's feedback and Finnish documents, and review them together. In this way, we can better understand it and distribute the translation work evenly. But this will take more time for the whole project, so it depends on the schedule and hours.

## **Risks**

RISK	ANALYSIS
Developers' lack of experience in used tools and methods	<p><b>Counter Measure:</b> Adequate time to let developers introduce themselves in used technology, even though most are skilled programmers.</p> <p><b>Analysis:</b> All of the team members learned necessary skills</p>
Quitting team members	<p><b>Counter Measure:</b> Good and friendly team atmosphere, motivating developers by changing tasks if needed and avoiding overburdening any developers. Managers also take part in developing if needed.</p> <p><b>Analysis:</b> No member of the group quitted during the project and a group spirit became even stronger towards the end of the project.</p>
Communication problems between team members	<p><b>Counter Measure:</b> Using a clear language, using different communication methods and tools effectively.</p> <p><b>Analysis:</b> There were no serious problems regarding the communication. And all team members were almost available via email and Skype.</p>
Managers' lack of experience in project management	<p><b>Counter Measure:</b> Prepare for management problem and spend more time for management issues.</p> <p><b>Analysis:</b> Managers had sometimes Skype meetings, and meetings with course staff were prepared well. As there were four managers, it was possible to share managing tasks between managers, but in many cases Otso was the only manager able to communicate with the client and discuss about problems.</p>
Communication problem with the client	<p><b>Counter Measure:</b> Since the client preferred to communicate in Finnish Language and we have only one manager who knows Finnish, we are trying to involve our supervisor and developers in the meeting with the client.</p> <p><b>Analysis:</b> Contacting the client is mostly done by two Finnish members in the group in addition to inviting supervisor to meetings.</p>
Changing the Requirements	<p><b>Counter Measure:</b> Use modular development model. Plan modules in a way</p>

	<p>they don't interfere each other. Be prepared for problems</p> <p><b>Analysis:</b>  Since the scope of the project was quite big, we tried to write all the necessary requirements and develop them according to their priority and project's estimated working hours. And we reached the minimum goals of the project.</p>
Lack of testing devices	<p><b>Counter Measure:</b>  Trying to get devices from the client, so no one will have to develop the project using Android Emulator.</p> <p><b>Analysis:</b>  The problem was quite marginal as not all team members developed Android projects in the start of the project. We got two Android smartphones and had them till the end of the project.</p>
Lack of testing	<p><b>Counter Measure:</b>  Introducing the team into Jenkins early enough.</p> <p><b>Analysis:</b>  Most of the team members don't have experience in testing. We managed to work on some testing cases and then focused more on what we have done and what needs to be complete.</p>
Different team members working different hours	<p>Even though most of the team members will get minimum working hours, the work was divided unequally as some members have got lot of hours in the late phase of the project and some others have worked much throughout the whole project. Some of the developers are getting maximum hours as some are getting minimum, but this was also predicted specially about the managers since we are four managers and three developers.</p> <p>Among the managers, Otso has got more working hours than other managers, as he has been the critical managers who could communicate with the client.</p> <p>Among the developers, Mikko has got more working hours than other developers, as he has the best knowledge of the project context and techniques.</p>

## Statistics

Team size	Dev. model	Start date	End data	Days	Hours
4+3	Modified Scrum	12.09.13	18.03.14	188	1049

*Table 1: General project information.*

Acitvity	Plannin g and manage ment	Req. specific ation.	Design	Code	Integrat ion and testing	Review s	Repair	Study	Other	Total
Hours	407	62	17	249	32	12.5	18	182	69.5	<b>1049</b>
%	38.8%	5.9%	1.6%	23.7%	3.1%	1.2%	1.7%	17.4%	6.6%	100%

*Table 2: Group effort by activity.*

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
66	10	0	7	1	12

*Table 3: Requirements and high-level design outcomes.*

Document	Pages	Versions
Preliminary analysis	7	5
Project Plan	20	10
Requirements specification	10	7
Final report	19	1
User interface document	2	9
Project's story	9	4
Project process flowchart	1	4
Database schema	1	5
System environment schema	1	1
Weekly reports	25	1
Test report	16	8
Review reports	6	1

*Table 4: Documents.*



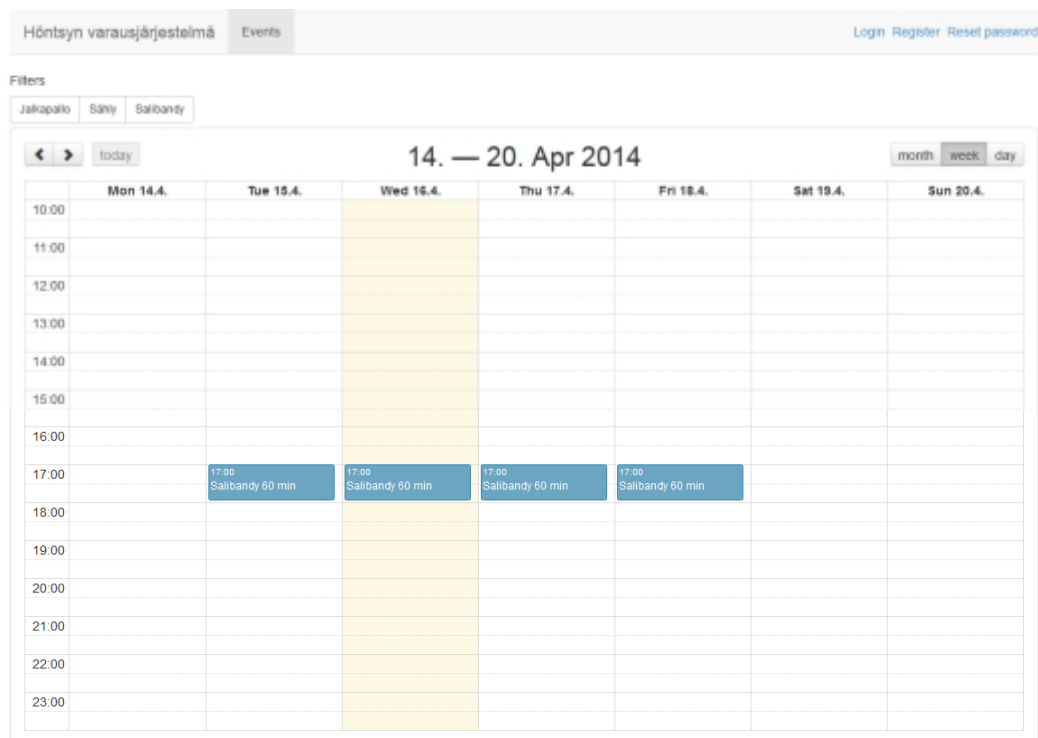
Language	Java
LOC	6255
Reused code	4%
Reused and modified	7%
Classes	26
Functions	129
Code revisions	157

*Table 5: Codelines.*

# Höntsyt

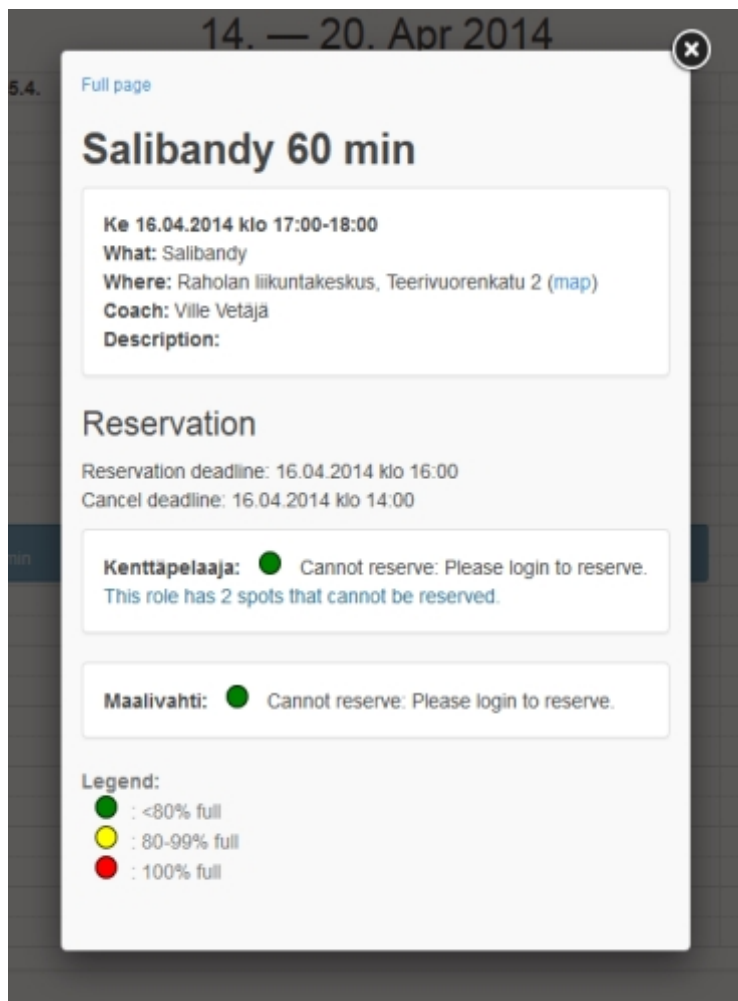
## Overview

The main idea of the project was to provide the client a reservation system to replace the one they had been previously using. The main goal for the team was to make a working and reliable application that the in the end of the project the client would be able to start using in their business. This reservation system was made to handle sports events that the client provides. The main feature of the system is a calendar like application where the administrator of the system can create, update and delete sporting events. Regular members can then sign up for these events or cancel their reservations. There's also managing system in place for the "coaches" of these events who mark people present to the system, mainly to track who has or has not been present in the said event. The system also includes further customization and management tools for the administrator.

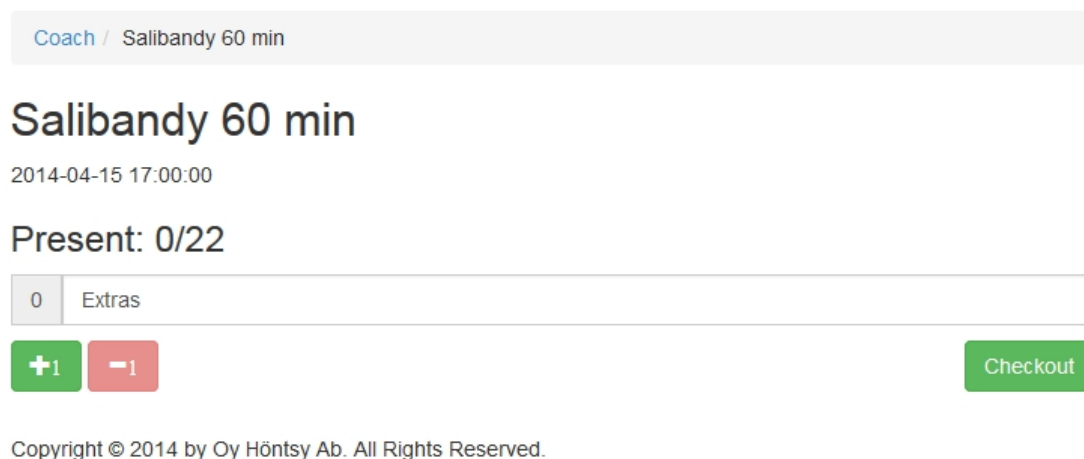


Copyright © 2014 by Oy Höntsyt Ab. All Rights Reserved.

Picture 1. Calendar view of the system.



Picture 2. iFrame pop-up for event sign up



Picture 3. Coach view of an empty event

## Organization and management

Starting the project the team consisted of three managers and four group members. However because of work-related issues one of the group members had to quit the course pretty much before it even started so for the rest of the project we had six people in the project.

Project Managers:

- Petri Arpiainen
- Teemu Ruotsalainen
- Ville Saarinen

Members

- Suvi Aho (Documentation and testing)
- Erkka Saarela (Programming)
- Julius Torkkeli (Testing and design)

The project staff was mainly divided to do things each of the members felt like they had the ability to do and if they had interest in particular field. In the end everyone found a field where they were most productive in. This however meant that the programming portion of the project fell basically on to the shoulders of one member which was rather unfortunate.

## Methods and tools

**IRC:** Went bit unused since the development work was mainly done by one person. Occasionally useful for quick chats.

**Subversion:** Code repository, high usage.

**Redmine:** Used to track working hours, tasks and more. Useful when everyone is committed to use it.

**Jenkins:** Didn't work with our selected framework (Yii) so went unused for the most part to not waste resources trying to get it work.

**Yii:** A PHP framework that we picked up since it seemed like a decent framework and no-one had any preferences or experience with other frameworks. Ended up working very well for our project.

**PHP:** Since our project was a web-application it was rather safe choice to use PHP as programming language.

**Modified SCRUM:** Daily meeting in this kind of projects aren't feasible so we used weekly meeting instead. Worked well enough.

**PostgreSQL:** For database use. Could have used MySQL as well if that was an option.

**Google Drive:** For document writing and sharing, immensely useful.

**Email:** Used for communication on important things that were not for quick chat on IRC.

**Various IDE and editors:** The developers used IDE and editors they were comfortable with.

## **Project phases and development model**

As mentioned above, our development method was modified SCRUM. We had the project divided in five phases (design, three implementation phases, documentation and finishing). Each of the implementation phases had plan for two sprints. However for the last phase we only had one sprint. The sprints lasted from two weeks at lowest and eight weeks at highest. Christmas break extended one of the sprints and few timetable issues extended the last one. We ended up working over the course limits too. This however meant that we managed to finish our projects goals fairly well.

**Project start and start of design phase:** 09.17.2013

**Preliminary analysis meeting:** 09.27.2013

**Project plan inspection, start of implementation (Sprint 1) and end of design phase:** 10.10.2013

**End of sprint 1 and start of sprint 2:** 10.30.2013

**First review, end of sprint 2 and start of sprint 3:** 11.20.2013

**Midway project presentation:** 12.04.2013

**Second review, end of sprint 3, start of sprint 4:** 12.13.2013

**End of sprint 4, start of sprint 5:** 01.15.2014

**Third review:** 01.28.2014

**Final presentation:** 05.03.2014

**End of sprint 5:** 14.03.2014

**Final meeting and project over:** 07.04.2014

## **Experiences**

The team didn't meet any unforeseen risks, which was great. However we met one of the risks at the start of the project when we lost one of the group members. Other risk that was really dominant through whole of the project was the amount of technical knowledge that the project member had, namely programming experience and skills. Also some lack of testing was an realized risk. This was mainly due to incompatibility between Yii-framework and Jenkins. Also for some time there were group members who weren't in touch with the group for some time, creating some anxiousness and uncertainty. Other than that there weren't many risks that did actually materialize during the project.

The project managers were long time friends between each other that really helped with communication between the managers. It also aided in creating positive atmosphere in the group as well. One word to describe the overall mood of the project would be "relaxed". We had code workshops at the beginning of the project where we tried to enhance our developers programming skills but managers lacked a true PHP-wizard to properly teach them. Despite all the unknowns through the project the group managed to develop a product that's ready for use and even more ready for further development.

## Statistics

Team size	Dev. model	Start date	End data	Days	Hours
3+3	SCRUM	09.13.2013	04.07.2014	206	918.25

*Table 1: General project information.*

Activity	Planning and management	Req. specific ation.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	351,75	4	20,25	199	68,75	12	31,25	157,25	74	<b>918.25</b>
%	38,31	0,44	2,21	21,67	7,49	1,31	3,4	17,12	8,06	100%

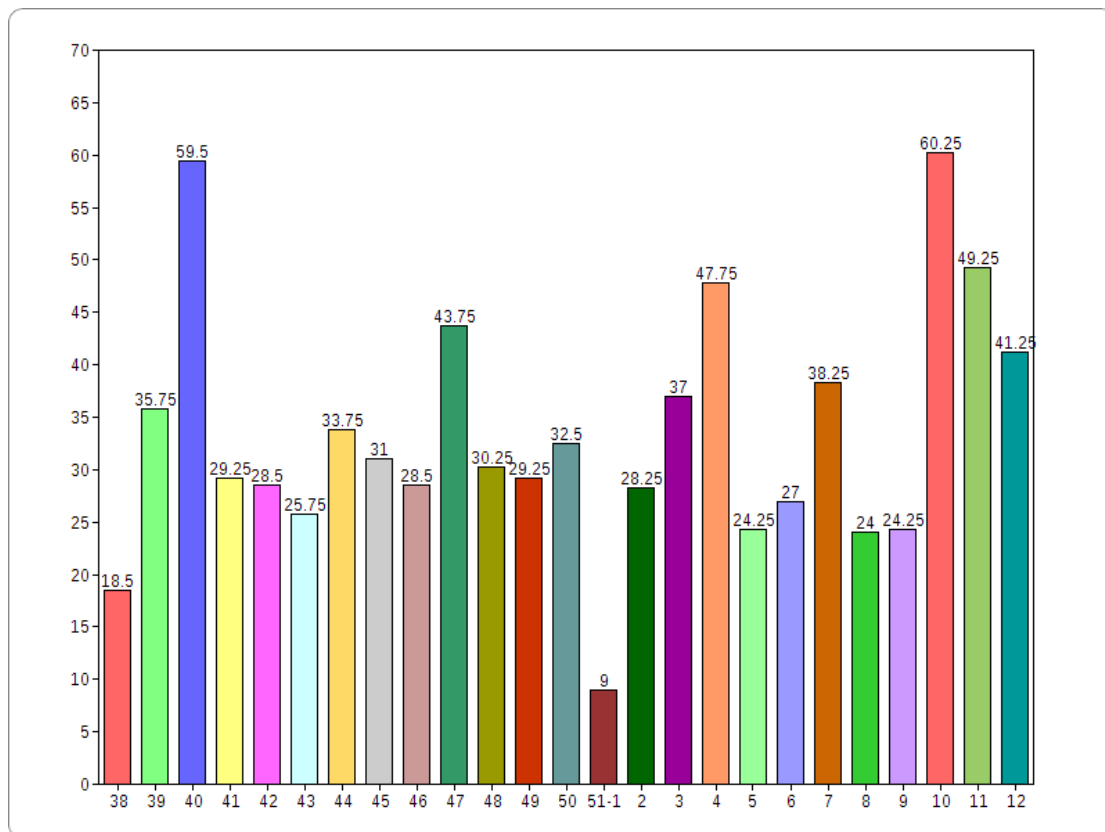
*Table 2: Group effort by activity.*

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
30	12	56	1	1	8

*Table 3: Requirements and high-level design outcomes.*

Document	Pages	Versions
Preliminary analysis	4	1
Project Plan	23	10
Requirements specification	12	5
Test plan	17	5
Test report	7	1
Final report	16	5
Project's story	6	1
Weekly reports	2	26

*Table 5: Documents.*



*Chart 1: Weekly working hours*